

epiworld

0.8.2

Generated by Doxygen 1.9.1

1 Example: 00-hello-world	1
2 Benchmarking	3
3 Contributor Code of Conduct	5
4 ModelMeaslesMixing	7
4.1 Overview	7
4.2 Disease States	7
4.3 Key Features	8
4.3.1 Disease Progression	8
4.3.2 Population Mixing	8
4.3.3 Public Health Measures	8
4.4 Parameters	8
4.4.1 Disease Parameters	8
4.4.2 Detection and Isolation	9
4.4.3 Contact Tracing and Quarantine	9
4.4.4 Vaccination	9
4.5 Usage Example	9
4.6 Differences from Other Models	10
4.6.1 vs. ModelMeaslesSchool	10
4.6.2 vs. ModelSEIRMixingQuarantine	10
4.7 Testing	10
5 epiworld c++ template library	11
5.1 Main features	11
5.2 Algorithm	11
5.3 Hello world (C++)	12
5.4 Surveillance simulation	12
5.4.1 Preliminary results	13
5.4.2 Cases detected	14
6 General parameters	15
6.1 Compartmental Models	15
6.1.1 SIR Model	15
6.1.2 SEIR Model	16
6.2 Agent-Based Model Approach	16
6.2.1 Mathematical preliminaries	17
6.2.2 Simulation study	17
6.3 Comparing ABM with Compartmental Models	17
6.3.1 SIR	17
6.3.2 SEIR	18
6.3.3 Rates	18

7 MIT License	19
8 model1	21
9 Mixing probabilities in connected model	23
9.1 Case 1: No grouping	23
9.2 Case 2: Grouping	24
10 EPI Simulator	25
10.1 Disease dynamics	25
10.2 Network dynamics	25
10.3 Contagion dynamics	25
10.4 Time dynamics	25
10.5 Updating agent's status	26
10.5.1 Other parameters	26
11 Namespace Index	27
11.1 Namespace List	27
12 Hierarchical Index	29
12.1 Class Hierarchy	29
13 Class Index	31
13.1 Class List	31
14 File Index	35
14.1 File List	35
15 Namespace Documentation	37
15.1 epiworld::sampler Namespace Reference	37
15.1.1 Detailed Description	37
15.1.2 Function Documentation	37
15.1.2.1 make_sample_virus_neighbors()	37
15.1.2.2 make_update_susceptible()	38
15.1.2.3 sample_virus_single()	38
15.2 sampler Namespace Reference	40
15.2.1 Detailed Description	40
15.2.2 Function Documentation	40
15.2.2.1 make_sample_virus_neighbors()	40
15.2.2.2 make_update_susceptible()	41
15.2.2.3 sample_virus_single()	41
16 Class Documentation	45
16.1 AdjList Class Reference	45
16.1.1 Constructor & Destructor Documentation	45
16.1.1.1 AdjList()	45

16.1.2 Member Function Documentation	46
16.1.2.1 read_edgelist()	46
16.2 epiworld::AdjList Class Reference	46
16.2.1 Constructor & Destructor Documentation	47
16.2.1.1 AdjList()	47
16.2.2 Member Function Documentation	47
16.2.2.1 read_edgelist()	47
16.3 Agent< TSeq > Class Template Reference	48
16.3.1 Detailed Description	50
16.3.2 Member Function Documentation	50
16.3.2.1 operator()()	50
16.3.2.2 swap_neighbors()	52
16.3.3 Friends And Related Function Documentation	52
16.3.3.1 default_rm_entity	52
16.4 epiworld::Agent< TSeq > Class Template Reference	53
16.4.1 Detailed Description	55
16.4.2 Member Function Documentation	55
16.4.2.1 operator()()	55
16.4.2.2 swap_neighbors()	56
16.4.3 Friends And Related Function Documentation	56
16.4.3.1 default_rm_entity	56
16.5 AgentsSample< TSeq > Class Template Reference	56
16.5.1 Detailed Description	57
16.5.2 Constructor & Destructor Documentation	57
16.5.2.1 AgentsSample()	57
16.6 epiworld::AgentsSample< TSeq > Class Template Reference	58
16.6.1 Detailed Description	58
16.6.2 Constructor & Destructor Documentation	59
16.6.2.1 AgentsSample()	59
16.7 DataBase< TSeq > Class Template Reference	59
16.7.1 Detailed Description	61
16.7.2 Member Function Documentation	62
16.7.2.1 get_generation_time()	62
16.7.2.2 get_reproductive_number()	62
16.7.2.3 get_transition_probability()	63
16.7.2.4 get_transmissions()	63
16.7.2.5 operator==([1/3])	63
16.7.2.6 operator==([2/3])	64
16.7.2.7 operator==([3/3])	64
16.7.2.8 record_virus()	64
16.8 epiworld::DataBase< TSeq > Class Template Reference	65
16.8.1 Detailed Description	67

16.8.2 Member Function Documentation	67
16.8.2.1 get_generation_time()	67
16.8.2.2 get_reproductive_number()	67
16.8.2.3 get_transition_probability()	68
16.8.2.4 get_transmissions()	68
16.8.2.5 operator==()	69
16.8.2.6 record_virus()	69
16.9 Entities< TSeq > Class Template Reference	69
16.9.1 Detailed Description	70
16.10 epiworld::Entities< TSeq > Class Template Reference	70
16.10.1 Detailed Description	71
16.11 Entities_const< TSeq > Class Template Reference	71
16.11.1 Detailed Description	71
16.12 epiworld::Entities_const< TSeq > Class Template Reference	72
16.12.1 Detailed Description	72
16.13 Entity< TSeq > Class Template Reference	73
16.13.1 Constructor & Destructor Documentation	73
16.13.1.1 Entity()	73
16.13.2 Friends And Related Function Documentation	74
16.13.2.1 default_rm_entity	74
16.14 epiworld::Entity< TSeq > Class Template Reference	74
16.14.1 Constructor & Destructor Documentation	75
16.14.1.1 Entity()	75
16.14.2 Friends And Related Function Documentation	75
16.14.2.1 default_rm_entity	76
16.15 epiworld::Event< TSeq > Struct Template Reference	76
16.15.1 Detailed Description	76
16.15.2 Constructor & Destructor Documentation	77
16.15.2.1 Event()	77
16.16 Event< TSeq > Struct Template Reference	77
16.16.1 Detailed Description	78
16.16.2 Constructor & Destructor Documentation	79
16.16.2.1 Event()	79
16.17 epiworld::GlobalEvent< TSeq > Class Template Reference	79
16.17.1 Detailed Description	80
16.17.2 Constructor & Destructor Documentation	80
16.17.2.1 GlobalEvent()	80
16.18 GlobalEvent< TSeq > Class Template Reference	81
16.18.1 Detailed Description	81
16.18.2 Constructor & Destructor Documentation	81
16.18.2.1 GlobalEvent()	81
16.19 epiworld::LFMCMC< TData > Class Template Reference	82

16.19.1 Detailed Description	83
16.20 LFMCMC< TData > Class Template Reference	83
16.20.1 Detailed Description	84
16.21 epiworld::Model< TSeq > Class Template Reference	85
16.21.1 Detailed Description	93
16.21.2 Member Function Documentation	93
16.21.2.1 add_globalevent()	93
16.21.2.2 clone_ptr()	94
16.21.2.3 draw()	94
16.21.2.4 events_add()	94
16.21.2.5 events_run()	95
16.21.2.6 load_agents_entities_ties()	95
16.21.2.7 reset()	96
16.21.2.8 run_multiple()	96
16.21.2.9 set_agents_data()	96
16.21.2.10 set_name()	97
16.21.2.11 write_data()	97
16.21.3 Member Data Documentation	98
16.21.3.1 initial_states_fun	98
16.21.3.2 rbinomd	98
16.21.3.3 rexp	98
16.21.3.4 rgammad	98
16.21.3.5 rgeomd	99
16.21.3.6 rlognormald	99
16.21.3.7 rnbino	99
16.21.3.8 rnormd	99
16.21.3.9 rpoissd	99
16.21.3.10 runifd	100
16.21.3.11 time_elapsed	100
16.22 Model< TSeq > Class Template Reference	100
16.22.1 Detailed Description	108
16.22.2 Member Function Documentation	109
16.22.2.1 add_globalevent()	109
16.22.2.2 clone_ptr()	109
16.22.2.3 draw()	109
16.22.2.4 events_add()	110
16.22.2.5 events_run()	110
16.22.2.6 load_agents_entities_ties()	112
16.22.2.7 reset()	112
16.22.2.8 run_multiple()	113
16.22.2.9 set_agents_data()	113
16.22.2.10 set_name()	113

16.22.2.11 write_data()	114
16.22.3 Member Data Documentation	114
16.22.3.1 initial_states_fun	114
16.22.3.2 rbinomd	114
16.22.3.3 rexp	115
16.22.3.4 rgammad	115
16.22.3.5 rgeomd	115
16.22.3.6 rlognormald	115
16.22.3.7 rnbino	115
16.22.3.8 rnormd	116
16.22.3.9 rpoissd	116
16.22.3.10 runifd	116
16.22.3.11 time_elapsed	116
16.23 epiworld::ModelDiagram Class Reference	117
16.24 ModelDiagram Class Reference	117
16.25 epiworld::epimodels::ModelDiffNet< TSeq > Class Template Reference	117
16.25.1 Detailed Description	118
16.26 ModelDiffNet< TSeq > Class Template Reference	119
16.26.1 Detailed Description	120
16.27 epiworld::epimodels::ModelMeaslesMixing< TSeq > Class Template Reference	120
16.27.1 Detailed Description	122
16.27.2 Constructor & Destructor Documentation	123
16.27.2.1 ModelMeaslesMixing() [1/2]	123
16.27.2.2 ModelMeaslesMixing() [2/2]	125
16.27.3 Member Function Documentation	126
16.27.3.1 clone_ptr()	126
16.27.3.2 get_agent_quarantine_triggered()	126
16.27.3.3 get_contact_matrix()	127
16.27.3.4 get_isolation_willingness()	127
16.27.3.5 get_quarantine_willingness()	127
16.27.3.6 initial_states()	127
16.27.3.7 run()	128
16.27.3.8 set_contact_matrix()	128
16.28 ModelMeaslesMixing< TSeq > Class Template Reference	129
16.28.1 Detailed Description	131
16.28.2 Constructor & Destructor Documentation	132
16.28.2.1 ModelMeaslesMixing() [1/2]	132
16.28.2.2 ModelMeaslesMixing() [2/2]	133
16.28.3 Member Function Documentation	134
16.28.3.1 clone_ptr()	134
16.28.3.2 get_agent_quarantine_triggered()	135
16.28.3.3 get_contact_matrix()	135

16.28.3.4	get_isolation_willingness()	135
16.28.3.5	get_quarantine_willingness()	136
16.28.3.6	initial_states()	136
16.28.3.7	run()	136
16.28.3.8	set_contact_matrix()	137
16.29	epiworld::epimodels::ModelMeaslesSchool< TSeq > Class Template Reference	137
16.29.1	Detailed Description	139
16.29.2	Constructor & Destructor Documentation	139
16.29.2.1	ModelMeaslesSchool()	139
16.29.3	Member Function Documentation	140
16.29.3.1	clone_ptr()	140
16.29.3.2	quarantine_agents()	141
16.29.3.3	reset()	141
16.30	ModelMeaslesSchool< TSeq > Class Template Reference	142
16.30.1	Detailed Description	143
16.30.2	Constructor & Destructor Documentation	144
16.30.2.1	ModelMeaslesSchool()	144
16.30.3	Member Function Documentation	145
16.30.3.1	clone_ptr()	145
16.30.3.2	quarantine_agents()	145
16.30.3.3	reset()	146
16.31	epiworld::epimodels::ModelSEIR< TSeq > Class Template Reference	146
16.31.1	Detailed Description	147
16.31.2	Member Function Documentation	148
16.31.2.1	initial_states()	148
16.31.3	Member Data Documentation	148
16.31.3.1	update_exposed_seir	148
16.31.3.2	update_infected_seir	149
16.32	ModelSEIR< TSeq > Class Template Reference	149
16.32.1	Detailed Description	150
16.32.2	Member Function Documentation	150
16.32.2.1	initial_states()	151
16.32.3	Member Data Documentation	151
16.32.3.1	update_exposed_seir	151
16.32.3.2	update_infected_seir	151
16.33	epiworld::epimodels::ModelSEIRCONN< TSeq > Class Template Reference	152
16.33.1	Constructor & Destructor Documentation	153
16.33.1.1	ModelSEIRCONN()	153
16.33.2	Member Function Documentation	153
16.33.2.1	clone_ptr()	154
16.33.2.2	initial_states()	154
16.33.2.3	reset()	154

16.34 ModelSEIRCONN< TSeq > Class Template Reference	155
16.34.1 Constructor & Destructor Documentation	156
16.34.1.1 ModelSEIRCONN()	156
16.34.2 Member Function Documentation	156
16.34.2.1 clone_ptr()	157
16.34.2.2 initial_states()	157
16.34.2.3 reset()	157
16.35 epiworld::epimodels::ModelSEIRD< TSeq > Class Template Reference	158
16.35.1 Detailed Description	159
16.35.2 Constructor & Destructor Documentation	159
16.35.2.1 ModelSEIRD() [1/2]	159
16.35.2.2 ModelSEIRD() [2/2]	160
16.35.3 Member Data Documentation	160
16.35.3.1 update_exposed_seir	160
16.36 ModelSEIRD< TSeq > Class Template Reference	161
16.36.1 Detailed Description	162
16.36.2 Constructor & Destructor Documentation	162
16.36.2.1 ModelSEIRD() [1/2]	162
16.36.2.2 ModelSEIRD() [2/2]	163
16.36.3 Member Data Documentation	163
16.36.3.1 update_exposed_seir	163
16.37 epiworld::epimodels::ModelSEIRDCONN< TSeq > Class Template Reference	164
16.37.1 Constructor & Destructor Documentation	165
16.37.1.1 ModelSEIRDCONN()	165
16.37.2 Member Function Documentation	166
16.37.2.1 clone_ptr()	166
16.37.2.2 initial_states()	166
16.37.2.3 reset()	166
16.38 ModelSEIRDCONN< TSeq > Class Template Reference	167
16.38.1 Constructor & Destructor Documentation	168
16.38.1.1 ModelSEIRDCONN()	168
16.38.2 Member Function Documentation	169
16.38.2.1 clone_ptr()	169
16.38.2.2 initial_states()	169
16.38.2.3 reset()	170
16.39 epiworld::epimodels::ModelSEIRMixing< TSeq > Class Template Reference	170
16.39.1 Constructor & Destructor Documentation	171
16.39.1.1 ModelSEIRMixing() [1/2]	172
16.39.1.2 ModelSEIRMixing() [2/2]	172
16.39.2 Member Function Documentation	173
16.39.2.1 clone_ptr()	173
16.39.2.2 initial_states()	173

16.39.2.3 reset()	174
16.40 ModelSEIRMixing< TSeq > Class Template Reference	174
16.40.1 Constructor & Destructor Documentation	175
16.40.1.1 ModelSEIRMixing() [1/2]	175
16.40.1.2 ModelSEIRMixing() [2/2]	176
16.40.2 Member Function Documentation	176
16.40.2.1 clone_ptr()	177
16.40.2.2 initial_states()	177
16.40.2.3 reset()	177
16.41 epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq > Class Template Reference	178
16.41.1 Detailed Description	180
16.41.2 Constructor & Destructor Documentation	180
16.41.2.1 ModelSEIRMixingQuarantine() [1/2]	181
16.41.2.2 ModelSEIRMixingQuarantine() [2/2]	182
16.41.3 Member Function Documentation	183
16.41.3.1 clone_ptr()	183
16.41.3.2 get_agent_quarantine_triggered()	183
16.41.3.3 get_contact_matrix()	184
16.41.3.4 get_isolation_willingness()	184
16.41.3.5 get_quarantine_willingness()	184
16.41.3.6 initial_states()	184
16.41.3.7 run()	185
16.41.3.8 set_contact_matrix()	185
16.42 ModelSEIRMixingQuarantine< TSeq > Class Template Reference	186
16.42.1 Detailed Description	188
16.42.2 Constructor & Destructor Documentation	188
16.42.2.1 ModelSEIRMixingQuarantine() [1/2]	189
16.42.2.2 ModelSEIRMixingQuarantine() [2/2]	190
16.42.3 Member Function Documentation	191
16.42.3.1 clone_ptr()	191
16.42.3.2 get_agent_quarantine_triggered()	191
16.42.3.3 get_contact_matrix()	192
16.42.3.4 get_isolation_willingness()	192
16.42.3.5 get_quarantine_willingness()	192
16.42.3.6 initial_states()	192
16.42.3.7 run()	193
16.42.3.8 set_contact_matrix()	193
16.43 epiworld::epimodels::ModelSIR< TSeq > Class Template Reference	194
16.43.1 Detailed Description	195
16.43.2 Member Function Documentation	195
16.43.2.1 initial_states()	195
16.44 ModelSIR< TSeq > Class Template Reference	195

16.44.1 Detailed Description	196
16.44.2 Member Function Documentation	197
16.44.2.1 initial_states()	197
16.45 epiworld::epimodels::ModelSIRCONN< TSeq > Class Template Reference	198
16.45.1 Constructor & Destructor Documentation	199
16.45.1.1 ModelSIRCONN()	199
16.45.2 Member Function Documentation	199
16.45.2.1 clone_ptr()	200
16.45.2.2 get_n_infected()	200
16.45.2.3 initial_states()	200
16.45.2.4 reset()	200
16.46 ModelSIRCONN< TSeq > Class Template Reference	201
16.46.1 Constructor & Destructor Documentation	202
16.46.1.1 ModelSIRCONN()	202
16.46.2 Member Function Documentation	203
16.46.2.1 clone_ptr()	203
16.46.2.2 get_n_infected()	203
16.46.2.3 initial_states()	203
16.46.2.4 reset()	204
16.47 epiworld::epimodels::ModelSIRD< TSeq > Class Template Reference	204
16.47.1 Detailed Description	205
16.47.2 Constructor & Destructor Documentation	205
16.47.2.1 ModelSIRD()	206
16.47.3 Member Function Documentation	207
16.47.3.1 initial_states()	207
16.48 ModelSIRD< TSeq > Class Template Reference	207
16.48.1 Detailed Description	208
16.48.2 Constructor & Destructor Documentation	209
16.48.2.1 ModelSIRD()	209
16.48.3 Member Function Documentation	209
16.48.3.1 initial_states()	209
16.49 epiworld::epimodels::ModelSIRDCONN< TSeq > Class Template Reference	210
16.49.1 Constructor & Destructor Documentation	211
16.49.1.1 ModelSIRDCONN()	211
16.49.2 Member Function Documentation	212
16.49.2.1 clone_ptr()	212
16.49.2.2 reset()	212
16.50 ModelSIRDCONN< TSeq > Class Template Reference	213
16.50.1 Constructor & Destructor Documentation	214
16.50.1.1 ModelSIRDCONN()	214
16.50.2 Member Function Documentation	214
16.50.2.1 clone_ptr()	214

16.50.2.2 reset()	215
16.51 epiworld::epimodels::ModelSIRLogit< TSeq > Class Template Reference	215
16.51.1 Detailed Description	216
16.51.2 Constructor & Destructor Documentation	217
16.51.2.1 ModelSIRLogit()	217
16.51.3 Member Function Documentation	218
16.51.3.1 clone_ptr()	218
16.51.3.2 reset()	218
16.52 ModelSIRLogit< TSeq > Class Template Reference	219
16.52.1 Detailed Description	220
16.52.2 Constructor & Destructor Documentation	220
16.52.2.1 ModelSIRLogit()	220
16.52.3 Member Function Documentation	221
16.52.3.1 clone_ptr()	221
16.52.3.2 reset()	221
16.53 epiworld::epimodels::ModelSIRMixing< TSeq > Class Template Reference	222
16.53.1 Constructor & Destructor Documentation	223
16.53.1.1 ModelSIRMixing() [1/2]	223
16.53.1.2 ModelSIRMixing() [2/2]	224
16.53.2 Member Function Documentation	224
16.53.2.1 clone_ptr()	225
16.53.2.2 initial_states()	225
16.53.2.3 reset()	225
16.54 ModelSIRMixing< TSeq > Class Template Reference	226
16.54.1 Constructor & Destructor Documentation	227
16.54.1.1 ModelSIRMixing() [1/2]	227
16.54.1.2 ModelSIRMixing() [2/2]	228
16.54.2 Member Function Documentation	228
16.54.2.1 clone_ptr()	228
16.54.2.2 initial_states()	228
16.54.2.3 reset()	229
16.55 epiworld::epimodels::ModelSIS< TSeq > Class Template Reference	229
16.55.1 Detailed Description	231
16.56 ModelSIS< TSeq > Class Template Reference	232
16.56.1 Detailed Description	233
16.57 epiworld::epimodels::ModelSISD< TSeq > Class Template Reference	233
16.57.1 Detailed Description	234
16.58 ModelSISD< TSeq > Class Template Reference	235
16.58.1 Detailed Description	236
16.59 epiworld::epimodels::ModelSURV< TSeq > Class Template Reference	236
16.59.1 Member Function Documentation	238
16.59.1.1 reset()	238

16.60 ModelSURV< TSeq > Class Template Reference	239
16.60.1 Member Function Documentation	240
16.60.1.1 reset()	241
16.61 Network< Nettype, Nodetype, Edgetype > Class Template Reference	241
16.62 epiworld::PersonTools< TSeq > Class Template Reference	241
16.63 PersonTools< TSeq > Class Template Reference	242
16.64 epiworld::Progress Class Reference	242
16.64.1 Detailed Description	242
16.65 Progress Class Reference	242
16.65.1 Detailed Description	242
16.66 epiworld::Queue< TSeq > Class Template Reference	243
16.66.1 Detailed Description	243
16.67 Queue< TSeq > Class Template Reference	243
16.67.1 Detailed Description	244
16.68 RandGraph Class Reference	244
16.69 epiworld::SAMPLETYPE Class Reference	245
16.70 SAMPLETYPE Class Reference	245
16.71 epiworld::Tool< TSeq > Class Template Reference	245
16.71.1 Detailed Description	246
16.72 Tool< TSeq > Class Template Reference	246
16.72.1 Detailed Description	248
16.73 epiworld::ToolFunctions< TSeq > Class Template Reference	248
16.73.1 Detailed Description	248
16.74 ToolFunctions< TSeq > Class Template Reference	249
16.74.1 Detailed Description	249
16.75 epiworld::Tools< TSeq > Class Template Reference	249
16.75.1 Detailed Description	249
16.76 Tools< TSeq > Class Template Reference	250
16.76.1 Detailed Description	250
16.77 epiworld::Tools_const< TSeq > Class Template Reference	251
16.77.1 Detailed Description	251
16.78 Tools_const< TSeq > Class Template Reference	251
16.78.1 Detailed Description	252
16.79 epiworld::UserData< TSeq > Class Template Reference	252
16.79.1 Detailed Description	253
16.79.2 Constructor & Destructor Documentation	254
16.79.2.1 UserData()	254
16.80 UserData< TSeq > Class Template Reference	254
16.80.1 Detailed Description	255
16.80.2 Constructor & Destructor Documentation	255
16.80.2.1 UserData()	255
16.81 epiworld::vecHasher< T > Struct Template Reference	256

16.81.1 Detailed Description	256
16.82 vecHasher< T > Struct Template Reference	256
16.82.1 Detailed Description	256
16.83 epiworld::Virus< TSeq > Class Template Reference	257
16.83.1 Detailed Description	258
16.84 Virus< TSeq > Class Template Reference	259
16.84.1 Detailed Description	261
16.85 epiworld::Viruses< TSeq > Class Template Reference	261
16.85.1 Detailed Description	261
16.86 Viruses< TSeq > Class Template Reference	262
16.86.1 Detailed Description	262
16.87 epiworld::Viruses_const< TSeq > Class Template Reference	263
16.87.1 Detailed Description	263
16.88 Viruses_const< TSeq > Class Template Reference	263
16.88.1 Detailed Description	264
16.89 epiworld::VirusFunctions< TSeq > Class Template Reference	264
16.90 VirusFunctions< TSeq > Class Template Reference	265
17 File Documentation	267
17.1 include/epiworld/agent-meat-state.hpp File Reference	267
17.1.1 Detailed Description	268
17.2 include/epiworld/models/measlesmixing.hpp File Reference	269
17.2.1 Detailed Description	269
17.2.2 Macro Definition Documentation	269
17.2.2.1 GET_MODEL	270
17.2.2.2 SAMPLE_FROM_PROBS	270
17.3 include/epiworld/models/seirmixingquarantine.hpp File Reference	270
17.3.1 Detailed Description	271
17.3.2 Macro Definition Documentation	271
17.3.2.1 GET_MODEL	271
17.3.2.2 SAMPLE_FROM_PROBS	271
Index	273

Chapter 1

Example: 00-hello-world

Output from the program:

```
Running the model...
|||||
SIMULATION STUDY
Name of the model      : (none)
Population size        : 10000
Agents' data           : (none)
Number of entities     : 0
Days (duration)        : 100 (of 100)
Number of viruses      : 1
Last run elapsed t     : 17.00ms
Last run speed         : 56.33 million agents x day / second
Rewiring               : off
Global events:
  (none)
Virus(es):
  - covid 19
Tool(s):
  - vaccine
Model parameters:
  (none)
Distribution of the population at time 100:
  - (0) Susceptible : 9950 -> 0
  - (1) Exposed     : 50 -> 0
  - (2) Recovered   : 0 -> 9399
  - (3) Removed     : 0 -> 601
Transition Probabilities:
  - Susceptible 0.92 0.08 - -
  - Exposed - 0.85 0.14 0.01
  - Recovered - - 1.00 -
  - Removed - - - 1.00
```

Chapter 2

Benchmarking

Here we keep a list of scenarios where we compare epiworld with other ABM simulation engines. Although the comparison is made at the speed level, we also list features of capabilities and main differences between the engines.

Chapter 3

Contributor Code of Conduct

As contributors and maintainers of this project, we pledge to respect all people who contribute through reporting issues, posting feature requests, updating documentation, submitting pull requests or patches, and other activities.

We are committed to making participation in this project a harassment-free experience for everyone, regardless of level of experience, gender, gender identity and expression, sexual orientation, disability, personal appearance, body size, race, ethnicity, age, or religion.

Examples of unacceptable behavior by participants include the use of sexual language or imagery, derogatory comments or personal attacks, trolling, public or private harassment, insults, or other unprofessional conduct.

Project maintainers have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct. Project maintainers who do not follow the Code of Conduct may be removed from the project team.

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by opening an issue or contacting one or more of the project maintainers.

This Code of Conduct is adapted from the Contributor Covenant (<http://contributor-covenant.org>), version 1.0.0, available at <http://contributor-covenant.org/version/1/0/0/>

Chapter 4

ModelMeaslesMixing

A comprehensive epidemiological model that combines measles-specific disease progression with population mixing and quarantine measures.

4.1 Overview

`ModelMeaslesMixing` implements a measles transmission model that extends the SEIR framework with:

- **Measles-specific disease progression:** Susceptible → Exposed → Prodromal → Rash → Recovered
- **Population mixing:** Uses contact matrices to model heterogeneous mixing between population groups
- **Infectious period:** Agents are infectious during the Prodromal state
- **Detection and isolation:** Detection occurs during the Rash state when symptoms become visible
- **Contact tracing:** Comprehensive contact tracing with configurable success rates
- **Quarantine measures:** Multiple quarantine states for exposed contacts
- **Vaccination:** Vaccine distribution with configurable efficacy and recovery enhancement

4.2 Disease States

The model includes 13 distinct states:

1. **Susceptible** - Can become infected
2. **Exposed** - Infected but not yet infectious (incubation period)
3. **Prodromal** - Infectious individuals (replaces "Infected" in standard SEIR)
4. **Rash** - Non-infectious with visible symptoms (detection occurs here)
5. **Isolated** - Detected individuals in self-isolation
6. **Isolated Recovered** - Recovered individuals still in isolation

7. **Detected Hospitalized** - Hospitalized individuals who were contact-traced
8. **Quarantined Exposed** - Exposed individuals in quarantine
9. **Quarantined Susceptible** - Susceptible individuals in quarantine
10. **Quarantined Prodromal** - Prodromal individuals in quarantine
11. **Quarantined Recovered** - Recovered individuals in quarantine
12. **Hospitalized** - Individuals requiring hospital care
13. **Recovered** - Immune individuals

4.3 Key Features

4.3.1 Disease Progression

- **Incubation Period:** Time from exposure to becoming infectious
- **Prodromal Period:** Duration of infectiousness before rash appears
- **Rash Period:** Duration of visible symptoms (detection window)
- **Detection:** Probabilistic detection during rash period
- **Hospitalization:** Some individuals require hospitalization

4.3.2 Population Mixing

- **Contact Matrices:** Define mixing patterns between population groups
- **Heterogeneous Mixing:** Different contact rates between groups
- **Scalable:** Supports multiple population entities

4.3.3 Public Health Measures

- **Contact Tracing:** Trace contacts of detected individuals
- **Quarantine:** Quarantine exposed contacts with configurable compliance
- **Isolation:** Isolate detected cases with configurable willingness
- **Vaccination:** Reduce susceptibility and enhance recovery

4.4 Parameters

4.4.1 Disease Parameters

- `contact_rate`: Average number of contacts per day
- `transmission_rate`: Probability of transmission per contact
- `incubation_period`: Average incubation period (days)
- `prodromal_period`: Average prodromal period (days)
- `rash_period`: Average rash period (days)
- `hospitalization_rate`: Probability of hospitalization
- `hospitalization_period`: Average hospitalization duration (days)

4.4.2 Detection and Isolation

- `days_undetected`: Average time until detection during rash period
- `isolation_period`: Duration of isolation for detected cases
- `isolation_willingness`: Proportion willing to self-isolate

4.4.3 Contact Tracing and Quarantine

- `contact_tracing_success_rate`: Probability of successfully tracing a contact
- `contact_tracing_days_prior`: Number of days to trace back
- `quarantine_period`: Duration of quarantine for contacts
- `quarantine_willingness`: Proportion willing to quarantine

4.4.4 Vaccination

- `prop_vaccinated`: Proportion of population vaccinated
- `vax_efficacy`: Vaccine efficacy (reduction in susceptibility)
- `vax_reduction_recovery_rate`: Recovery enhancement from vaccination

4.5 Usage Example

```
#include "epiworld.hpp"
// Contact matrix for single homogeneous population
std::vector<double> contact_matrix = {1.0};
// Create the model
epimodels::ModelMeaslesMixing<> model(
    "Measles",           // Virus name
    1000,                // Population size
    0.005,               // Initial prevalence
    4.0,                 // Contact rate
    0.9,                 // Transmission rate
    0.95,                // Vaccine efficacy
    0.5,                 // Vaccine recovery enhancement
    10.0,                // Incubation period
    4.0,                 // Prodromal period
    5.0,                 // Rash period
    contact_matrix,      // Contact matrix
    0.1,                 // Hospitalization rate
    7.0,                 // Hospitalization period
    2.0,                 // Days undetected
    14,                  // Quarantine period
    0.8,                 // Quarantine willingness
    0.9,                 // Isolation willingness
    10,                  // Isolation period
    0.7,                 // Vaccination rate
    0.8,                 // Contact tracing success rate
    3u,                  // Contact tracing days prior
);
// Add population entity
model.add_entity(Entity<>("Population", dist_factory<>(0, 1000)));
// Run simulation
model.run(60, 123);
model.print();
```

4.6 Differences from Other Models

4.6.1 vs. ModelMeaslesSchool

- **Population Mixing:** Adds contact matrices for heterogeneous mixing
- **Contact Tracing:** Enhanced contact tracing with configurable success rates
- **Scalability:** Supports multiple population entities

4.6.2 vs. ModelSEIRMixingQuarantine

- **Disease-Specific States:** Uses Prodromal/Rash instead of generic Infected
- **Detection Timing:** Detection occurs during Rash state, not Infected state
- **Infectious Period:** Only Prodromal individuals are infectious
- **Vaccination:** Includes vaccine distribution and efficacy

4.7 Testing

The model includes comprehensive tests that verify:

- Correct transition probabilities between states
- Proper detection and quarantine mechanics
- Population mixing functionality
- Reproductive number calculations
- All quarantine states function correctly

See `tests/19-measles-mixing.cpp` for test examples.

Chapter 5

epiworld c++ template library

5.1 Main features

This C++ template-header-only library provides a general framework for epidemiologic simulation. The main features of the library are:

1. Four key classes: `Model`, `Person`, `Tool`, and `Virus`.
2. The model features a social networks of `Persons`.
3. `Persons` can have multiple `Tools` as a defense system.
4. `Tools` can reduce contagion rate, transmissibility, death rates, and improve recovery rates.
5. `Viruses` can mutate (generating new variants).
6. `Models` can feature multiple states, e.g., `HEALTHY`, `SUSCEPTIBLE`, etc.
7. `Models` can have an arbitrary number of parameters.
8. **REALLY FAST** About 6.5 Million person/day simulations per second.

5.2 Algorithm

Setup

- Create viruses.
- Create tools (arbitrary).
- Set model parameters (arbitrary).
- Create global events (e.g., surveillance).
- Set up the population: small world network (default).
- Set up rewiring (optional).
- Set states (arbitrary number of them).

Run

1. Distribute the tool(s) and virus(es)
2. For each t in 1 -> Duration:
 - Update state for susceptible/infected/removed(?)
 - Mutate virus(es) (each individual)
 - Run Global events (e.g., surveillance)
 - Run rewiring algorithm

Along update:

- Contagion events are applied recorded.
- New variants are recorded.
- Optional user data is recorded.

5.3 Hello world (C++)

```
#include "include/epiworld/epiworld.hpp"
int main()
{
    // Creating a virus
    epiworld::Virus<> covid19("covid 19", .01, true);
    covid19.set_infectiousness(.8);

    // Creating a tool
    epiworld::Tool<> vax("vaccine", .5, true);
    vax.set_contagion_reduction(.95);
    // Creating a model
    epiworld::Model<> model;
    // Adding the tool and virus
    model.add_virus(covid19);
    model.add_tool(vax);
    // Generating a random pop
    model.population_from_adjlist(
        epiworld::rgraph_smallworld(1000, 5, .2)
    );
    // Initializing setting days and seed
    model.init(60, 123123);
    // Running the model
    model.run();
    model.print();
    return;
}
```

5.4 Surveillance simulation

- Incubation time of the disease $\sim \text{Gamma}(3, 1)$
- Duration of the disease $\sim \text{Gamma}(12, 1)$
- Probability of becoming symptomatic: 0.9
- Prob. of transmission: 1.0.
- Vaccinated population: 25%
- Vaccine efficacy: .9.
- Vaccine reduction on transmission: 0.5.
- Surveillance program of x% of the population at random.
- Individuals who test positive become isolated.

5.4.1 Preliminary results

```
# With low surveillance
pop_size <- 20e3
pop_seed <- pop_size * .01
s_levels <- c(0.0001, 0.002)
system(sprintf("./07-surveillance.o %i %i 100 %.04f 2>&1", pop_seed, pop_size, s_levels[1]), intern = TRUE)
|>
cat(sep = "\n")

## Running the model...
##
## | done.
##
##
## SIMULATION STUDY
##
## Population size      : 20000
## Days (duration)     : 200 (of 200)
## Number of variants  : 1
## Last run elapsed t   : 505.00ms
## Rewiring            : off
##
## Virus(es):
## - Covid19 (baseline prevalence: 100 seeds)
## Tool(s):
## - Vaccine (baseline prevalence: 25.00%)
##
## Model parameters:
## - Infect period      : 12.0000
## - Latent period      : 3.0000
## - Prob of symptoms   : 0.7000
## - Prob of transmission : 1.0000
## - Prob. death        : 0.0010
## - Prob. reinfect     : 0.1000
## - Surveillance prob. : 1.0e-04
## - Vax efficacy       : 0.9000
## - Vax redux transmission : 0.5000
##
## Distribution of the population at time 200:
## - Total susceptible (S) : 19900 -> 2106
## - Total recovered (S)   : 0 -> 17369
## - Total latent (I)      : 100 -> 109
## - Total symptomatic (I) : 0 -> 155
## - Total symptomatic isolated (I) : 0 -> 2
## - Total asymptomatic (I) : 0 -> 72
## - Total asymptomatic isolated (I) : 0 -> 0
## - Total removed (R)    : 0 -> 187
##
## (S): Susceptible, (I): Infected, (R): Recovered
##
hist1 <- read.csv("07-surveillance_hist.txt", sep = " ")
surv1 <- read.csv("07-surveillance_user_data.txt", sep = " ")
# With high surveillance
system(sprintf("./07-surveillance.o %i %i 100 %.04f 2>&1", pop_seed, pop_size, s_levels[2]), intern = TRUE)
|>
cat(sep = "\n")

## Running the model...
##
## | done.
##
##
## SIMULATION STUDY
##
## Population size      : 20000
## Days (duration)     : 200 (of 200)
## Number of variants  : 1
## Last run elapsed t   : 530.00ms
## Rewiring            : off
##
## Virus(es):
```

```
## - Covid19 (baseline prevalence: 100 seeds)
## Tool(s):
## - Vaccine (baseline prevalence: 25.00%)
##
## Model parameters:
## - Infect period      : 12.0000
## - Latent period      : 3.0000
## - Prob of symptoms   : 0.7000
## - Prob of transmission : 1.0000
## - Prob. death        : 0.0010
## - Prob. reinfect     : 0.1000
## - Surveillance prob. : 0.0020
## - Vax efficacy       : 0.9000
## - Vax redux transmission : 0.5000
##
## Distribution of the population at time 200:
## - Total susceptible (S)      : 19900 -> 2125
## - Total recovered (S)       : 0 -> 17325
## - Total latent (I)          : 100 -> 109
## - Total symptomatic (I)     : 0 -> 155
## - Total symptomatic isolated (I) : 0 -> 8
## - Total asymptomatic (I)    : 0 -> 76
## - Total asymptomatic isolated (I) : 0 -> 1
## - Total removed (R)        : 0 -> 201
##
## (S): Susceptible, (I): Infected, (R): Recovered
##
hist2 <- read.csv("07-surveillance_hist.txt", sep = " ")
surv2 <- read.csv("07-surveillance_user_data.txt", sep = " ")
hist_comb <- rbind(
  cbind(sim = as.character(s_levels[1]), hist1),
  cbind(sim = as.character(s_levels[2]), hist2)
)
ggplot(hist_comb, aes(x = date, y = counts + 1, colour = state, linetype=sim)) +
  geom_line() +
  # scale_y_log10() +
  labs(y = "Counts (log)")
```

5.4.2 Cases detected

```
survdat <- rbind(
  with(surv1, rbind(
    data.frame(Id = as.character(s_levels[1]), Date = date, Type = "N Sampled", n = nsampled),
    data.frame(Id = as.character(s_levels[1]), Date = date, Type = "N detected", n = ndetected),
    data.frame(Id = as.character(s_levels[1]), Date = date, Type = "N detected Asymp", n =
      ndetected_asymp),
    data.frame(Id = as.character(s_levels[1]), Date = date, Type = "N Asymp", n = nasymptomatic)
  )),
  with(surv2, rbind(
    data.frame(Id = as.character(s_levels[2]), Date = date, Type = "N Sampled", n = nsampled),
    data.frame(Id = as.character(s_levels[2]), Date = date, Type = "N detected", n = ndetected),
    data.frame(Id = as.character(s_levels[2]), Date = date, Type = "N detected Asymp", n =
      ndetected_asymp),
    data.frame(Id = as.character(s_levels[2]), Date = date, Type = "N Asymp", n = nasymptomatic)
  ))
)
ggplot(survdat, aes(x = Date, y = n + 1, colour = Type)) +
  geom_line() +
  facet_wrap(~Id) +
  scale_y_log10() +
  labs(y = "Counts (log)")
```

Chapter 6

General parameters

The following are parameters used for both ABM and Compartmental models.

```
EPI_BETA <- 0.75
EPI_GAMMA <- 0.33
EPI_LATENCY <- 1/0.33
EPI_N <- 10000
EPI_0 <- 0.01
EPI_NDAYS <- 50
Sys.setenv( # nolint
  EPI_BETA = EPI_BETA,
  EPI_GAMMA = EPI_GAMMA,
  EPI_LATENCY = EPI_LATENCY,
  EPI_N = EPI_N,
  EPI_0 = EPI_0,
  EPI_NDAYS = EPI_NDAYS
)
```

6.1 Compartmental Models

6.1.1 SIR Model

```
library(deSolve)
library(ggplot2)
library(data.table)
# Code from
# Chapter 2: SIR
# Book "Epidemics: Models and Data using R."
# By: Ottar N. Bjørnstad
sirmod <- function(t, y, parms) {
  # Pull state variables from the vector y
  S = y[1]
  I = y[2]
  R = y[3]

  # Pull parameter values from parms vector
  beta = parms["beta"]
  mu = parms["mu"]
  gamma = parms["gamma"]
  N = parms["N"]

  # Define equations
  dS = mu * (N - S) - beta * S * I/N
  dI = beta * S * I/N - (mu + gamma) * I
  dR = gamma * I - mu * R
  res = c(dS, dI, dR)

  # Return list of gradients
  list(res)
}
# Initial parameters
times <- seq(0, EPI_NDAYS, by = 1)
parms <- c(mu = 0, N = EPI_N, beta = EPI_BETA, gamma = EPI_GAMMA)
start <- c(S = EPI_N * (1 - EPI_0), I = EPI_N * EPI_0, R = 0)
out <- ode(y = start, times = times, func = sirmod, parms = parms)
out <- as.data.frame(out)
out <- rbind(
```

```

with(out, data.table(date = time, state = "Susceptible", counts = S)),
with(out, data.table(date = time, state = "Infected", counts = I)),
with(out, data.table(date = time, state = "Recovered", counts = R))
)

```

Now we visualize the model

```

ggplot(out, aes(x = date, y = counts)) +
  geom_line(aes(colour = state)) +
  labs(title = "Compartmental SIR")

```

6.1.2 SEIR Model

```

# Code adapted from
# Chapter 2: SIR
# Book "Epidemics: Models and Data using R"
# By: Ottar N. Bjørnstad
seirmod <- function(t, y, parms) {
  # Pull state variables from y vector
  S = y[1]
  E = y[2]
  I = y[3]
  R = y[4]

  # Pull parameter values from parms vector
  beta = parms["beta"]
  mu = parms["mu"]
  alpha = parms["alpha"]
  gamma = parms["gamma"]
  N = parms["N"]

  # Define equations
  dS = mu * (N - S) - beta * S * I/N - mu * S
  dE = beta * S * I/N - (mu + alpha) * E
  dI = alpha * E - (mu + gamma) * I
  dR = gamma * I - mu * R
  res = c(dS, dE, dI, dR)

  # Return list of gradients
  list(res)
}

# Initial parameters
parms <- c(
  mu = 0, N = EPI_N, beta = EPI_BETA,
  alpha = 1/EPI_LATENCY, gamma = EPI_GAMMA
)

start <- c(S = EPI_N * (1 - EPI_0), E = EPI_N * EPI_0, I = 0, R = 0)
out_seir <- ode(y = start, times = times, func = seirmod, parms = parms)
out_seir <- as.data.frame(out_seir)
out_seir <- rbind(
  with(out_seir, data.table(date = time, state = "Susceptible", counts = S)),
  with(out_seir, data.table(date = time, state = "Exposed", counts = E)),
  with(out_seir, data.table(date = time, state = "Infected", counts = I)),
  with(out_seir, data.table(date = time, state = "Recovered", counts = R))
)

```

Now we visualize the model

```

ggplot(out_seir, aes(x = date, y = counts)) +
  geom_line(aes(colour = state)) +
  labs(title = "Compartmental SEIR")

```

6.2 Agent-Based Model Approach

Calculation of the expected number of days in state S when prob of changing state equals α is $1/\alpha$

```

set.seed(712)
a <- .3
R <- matrix(runif(2e5 * 50), ncol = 50)
dat <- apply(R, 1, \ (x) {
  which.max(x < a)
})
mean(dat) - 1 / a

```

```
[1] -0.01049333
```


6.2.1 Mathematical preliminaries

That agent i becomes infected can be computed as follows:

At the same time, the probability of not becoming infected equals to the probability of no infected agent transmitting the infection. The probability that agent j infects i equals

In this case, β is parametrized such that its values are within $(0,1)$. Since transmission from the i infected agents happens independently, we finally have the following:

With the above equation, we can now calculate the change in the number of susceptible agents. In this case, it equals the expected number of new infections:

With the same parametrization in the canonical SIR model (Kermack and McKendrick), the instantaneous change in the number of susceptible agents equals $\frac{dS}{dt} = -S \beta I$. Given S and I , we can show that, as $\beta \rightarrow 0$, i.e., the population grows, both rates converge to the same number. Formally:

The same can be shown for the change in the number recovered.

6.2.2 Simulation study

Now, what happens with `epiworld`.

```
system("./09-sir-connected.o -n $EPI_N -b $EPI_BETA -d $EPI_NDAYS -p $EPI_0 -r $EPI_GAMMA -i 1.0 -s555599")
library(ggplot2)
epiworld <- data.table::fread("total_hist.txt")
ggplot(epiworld, aes(x = date, y = counts)) +
  geom_line(aes(colour = state)) +
  labs(title = "ABM SIR")
system("./09-seir-connected.o -n $EPI_N -b $EPI_BETA -d $EPI_NDAYS -p $EPI_0 -r $EPI_GAMMA -i 1.0 -s555599
-l $EPI_LATENCY")
library(ggplot2)
epiworld <- data.table::fread("total_hist.txt")
ggplot(epiworld, aes(x = date, y = counts)) +
  geom_line(aes(colour = state)) +
  labs(title = "ABM SEIR")
```

6.3 Comparing ABM with Compartmental Models

To this end, we will compare the results of the first run of the Compartmental model with 100 runs of the ABM, compute the confidence interval, and see how likely is the compartmental model to fall within the trajectory of the ABM simulation.

6.3.1 SIR

```
system("./09-sir-connected.o -n $EPI_N -b $EPI_BETA -d $EPI_NDAYS -p $EPI_0 -r $EPI_GAMMA -i 1.0 -s555599 -e
100")
library(ggplot2)
library(data.table)
epiworld <- data.table::fread("09-sir-connected-experiments.csv")
epiworld <- epiworld[, .(
  min = quantile(counts, probs = .025),
  mean = mean(counts),
  max = quantile(counts, probs = .975)), by = .(date, state)]
# Merging Compartmental
epiworld <- merge(
  epiworld,
  out[, .(date = date, state = state, compartmental = counts)],
  by = c("date", "state")
)
setorder(epiworld, state, date)
ggplot(epiworld, aes(x = date, y = mean)) +
  geom_ribbon(aes(ymin = min, ymax = max, colour = state), alpha = 0.1) +
  geom_line(aes(x = date, y = compartmental, colour = sprintf("%s (compt)", state)))
```

It seems that, although both yield the same equilibria, compartmental models reach the highest point of the simulation earlier. This makes sense as within a single day of the ABM simulation, compartmental models have more events taking place. Nonetheless, as predicted, as $\beta \rightarrow 0$, the differences become lesser. Furthermore, we could use the fact that the transition rates are known to compute an adjustment.

6.3.2 SEIR

```
system("./09-seir-connected.o -n $EPI_N -b $EPI_BETA -d $EPI_NDAYS -p $EPI_0 -r $EPI_GAMMA -i 1.0 -s555599
-e 100 -l $EPI_LATENCY")
library(ggplot2)
library(data.table)
epiworld_seir <- data.table::fread("09-seir-connected-experiments.csv")
epiworld_seir <- epiworld_seir[, .(
  min = quantile(counts, probs = .025),
  mean = mean(counts),
  max = quantile(counts, probs = .975)), by = .(date, state)]
# Merging Compartmental
epiworld_seir <- merge(
  epiworld_seir,
  out_seir[, .(date = date, state = state, compartmental = counts)],
  by = c("date", "state")
)
setorder(epiworld_seir, state, date)
ggplot(epiworld_seir, aes(x = date, y = mean)) +
  geom_ribbon(aes(ymin = min, ymax = max, colour = state), alpha = 0.1) +
  geom_line(aes(x = date, y = compartmental, colour = sprintf("%s (compt)", state)))
```

6.3.3 Rates

```
S <- 1000
rate_comp <- function(I,B) S * B * I
rate_abm <- function(I,B) S * (1 - (1 - B)^I)
op <- par(mfrow = c(3, 2))
for (i in c(1, 10, 100)) {
  curve(rate_comp(i, x), from = .01, to = 0.05)
  curve(rate_abm(i, x), from = .01, to = 0.05, add = FALSE, lty = 2)
}
par(op)
```

Chapter 7

MIT License

Copyright (c) 2021 George G. Vega Yon

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Chapter 8

model1

The dynamics of the simulation process are:

1. Discrete Markov process.
2. The simulation has the following parameters:
 - a. New variant emergence at rate X .
 - b. For each variant k :
 - Unvaccinated individuals become sick rate $C(k)$,
 - Mortality rate $D(k)$,
 - Recovery rate $H(k)$,
 - Vaccines have an efficacy rate $E(v, k)$ and pseudo vaccines (recovered) have efficacy rate $E(r, k) < E(v, k)$. In general, the probability of i acquiring the disease k from j will be equal to

$$P(i \text{ gets the disease from } j \mid \text{their states}) = C(k) * (1 - E(i, k)) * (1 - E(j, k))$$

where $(i, j) \in (u, v, r)$. Efficacy rate for unvaccinated is zero.

- Vaccinated individuals have a reduced mortality rate $D(k, v) > D(k)$, and recovered individuals $D(k, r) \in (D(k, v), D(k))$
- Vaccinated individuals have an increased recovery rate $H(k, v) > H(k)$, whereas recovered's rate $H(k, r) \in (H(k), H(k, v))$.

The sum of mortality and recovery rates is less than one since the difference represents no change.

- c. Each country vaccinates citizens at rate V function of A (availability) and B (citizens' acceptance rate.)
- d. In each country i , the entire population $N(i)$ distributes between the following states:

- Healthy unvaccinated ($N(i, t, u)$),
- Healthy vaccinated ($N(i, t, v)$),
- Deceased ($N(i, t, d)$),
- Recovered ($N(i, t, r)$),
- Unvaccinated and sick with variant ($N(i, t, s, k|u)$) k ., and
- Vaccinated and sick with variant ($N(i, t, s, k|v)$) k .

$$\text{Total sick are } N(i, t, k, s) = \sum_{g \in \{u, v\}} N(i, t, k, s|g)$$

Globally, we keep track of the prevalence of new variants. Variants can disappear if no more individuals port the variant, i.e., the prevalence rate $P(k, t) = \sum_i N(i, s, k)$ equals zero.

- d. Vaccines are manufactured at each country at rates $M(i)$ and uniformly shared with other countries at rate $S(i)$.
- c. Population flows between each country pair (i, j) at a rate $F(i, j)$. Flows between countries do not change Population and are symmetric.

3. The simulation process is as follows:

- (a) Countries are initialized with a total population $N(i)$.
- (b) Variant zero initializes at a random location i , with an initial prevalence $P(k, t) = N(i, t, k)$.
- (c) For time t in $(0, T)$ do:
 - a. Unvaccinated individuals can become sick of variant k with probability:

$$\Pr(h \rightarrow s | i, t, k, u) \sim \sum(g \in \{u, v\}) (N(i, t-1, s, k | g) + \sum(j \neq i) F(i, j) * N(j, t-1, s, k | g)) * C(k) / (N(i) + \sum(j \neq i) N(j))$$
 - b. Vaccinated individuals can become sick of variant k with probability: $\Pr(v \rightarrow s | i, t, k, v) \sim \Pr(h \rightarrow s | i, t, k) * (1 - E(v, k))$.
 - b. Recovered individuals can become sick of variant k with probability: $\Pr(v \rightarrow s | i, t, k, r) \sim \Pr(h \rightarrow s | i, t, k) * (1 - E(r, k))$.
 - c. Sick individuals with variant k die with probability $D(k)$ or recover with probability $H(k)$, otherwise they stay infected; with the rates depending on their vaccination status v or n .
 - d. Unvaccinated individuals vaccinate in country i with probability $P(u \rightarrow v) \sim V(A(i, t), B(i))$.
 - e. The country vaccine supply changes.

Chapter 9

Mixing probabilities in connected model

George G. Vega Yon, Ph.D. 2024-04-25

9.1 Case 1: No grouping

We will look into the probability of drawing infected individuals to simplify the algorithm. There are I infected individuals at any time in the simulation; thus, instead of drawing from $\text{Bern}(c/N, N)$, we will be drawing from $\text{Bern}(c/N, I)$. The next step is to check which infected individuals should be drawn. Let's compare the distributions using the hypergeometric as an example:

```
set.seed(132)
nsims <- 1e4
N <- 400
rate <- 5
p <- rate/N
I <- 10
sim_complex <- parallel::mclapply(1:nsims, \(i) {
  nsamples <- rbinom(N, N, p)
  sum(rbinom(N, size = nsamples, prob = I/N) > 0)
}, mc.cores = 4L) |> unlist()
sim_simple <- parallel::mclapply(1:nsims, \(i) {
  sum(rbinom(N, I, p) > 0)
}, mc.cores = 4L) |> unlist()
op <- par(mfrow = c(1,2))
MASS::truehist(sim_complex)
MASS::truehist(sim_simple)
par(op)
quantile(sim_complex)
```

0%	25%	50%	75%	100%
27	43	47	51	71

```
quantile(sim_simple)
```

0%	25%	50%	75%	100%
23	43	47	51	71

```
plotter(sim_complex, sim_simple)
```

These two approaches are equivalent, but the second one is more efficient from the computational perspective.

9.2 Case 2: Grouping

This explores the case when we have mixing across groups. The question is if we can replicate the effect at the group level.

```
set.seed(123133)
ngroups <- 3
mixing <- matrix(
  c(0.1, 0.2, 0.3, 0.2, 0.1, 0.2, 0.3, 0.2, 0.1),
  nrow = ngroups,
  ncol = ngroups
)
mixing <- mixing/rowSums(mixing)
mixing

      [,1]      [,2]      [,3]
[1,] 0.1666667 0.3333333 0.5000000
[2,] 0.4000000 0.2000000 0.4000000
[3,] 0.5000000 0.3333333 0.1666667

N <- 500
sizes <- c(100, 150, 250)
rate <- 5
p <- rate/N
I <- c(10, 30, 20)
ids <- rep.int(1:ngroups, times = sizes)
nsims <- 1e4
sim_complex <- parallel::mclapply(1:nsims, \(i) {
  # Sampling group first
  sapply(1:ngroups, \(g) {
    # How many each individual will sample from the groups
    ans <- rbinom(
      n = N, size = sizes[g], prob = mixing[ids,][g] * p
    ) |> sum()
    # Sampling with replacement
    rbinom(ans, size = 1, prob = I[g]/sizes[g]) |> sum()
  }) |> sum()
}, mc.cores = 4L) |> unlist()
```

Using the alternative method in which we directly weight the probabilities:

```
sim_simple <- parallel::mclapply(1:nsims, \(i) {
  # Sampling group first
  sapply(1:ngroups, \(g) {
    rbinom(
      n = N, size = I[g], prob = mixing[cbind(ids,g)] * p
    ) |> sum()
  }) |> sum()
}, mc.cores = 4L) |> unlist()
op <- par(mfrow = c(1,2))
MASS::truehist(sim_complex)
MASS::truehist(sim_simple)
par(op)
quantile(sim_complex)
```

```
0% 25% 50% 75% 100%
57  88  94 101 131
```

```
quantile(sim_simple)
```

```
0% 25% 50% 75% 100%
58  87  94 101 135
```

```
plotter(sim_complex, sim_simple)
```


Chapter 10

EPI Simulator

10.1 Disease dynamics

Diseases continuously evolve in time. Changes in their genetic sequence make them more or less resistant to the particular version of the vaccine. Mutations also affect the transmissibility level and mortality rate of the disease. Using this approach allows making vaccination efficacy a function of compatibility between the variant and the vaccine.

When an individual becomes infected, the disease accumulates mutations in the new host. Ultimately, there is no single version of the disease present in the model, but rather an infinite number of them, each slightly different from the other.

10.2 Network dynamics

We can assume that the Population is organized in fully connected blocks for the first version of the model. Block sizes and the number of connections between blocks are Poisson random variables. Individuals interact with all the members of their blocks, and bridging individuals allow the disease to move across blocks.

10.3 Contagion dynamics

The transmission of the disease will be governed by the number of vaccinated, infected, and recovered within each block. Transmission between blocks will be treated in the same way, although individuals bridging the block will only interact with others within the block and their direct connections across the blocks.

10.4 Time dynamics

Time dynamics has two components, how biology evolves and how agents react.

The model develops as a continuous-time Markov process. Each block of individuals takes action at rates $L(i|N(i))$ function of the local number of infections. This way, if

10.5 Updating agent's status

Like most other components, updating agents' states can be personalized. A naive approach allows agents to get infected with a single virus or stay as-is. The probability of this event is conditional on acquiring at most one virus. Since these are independent events, the conditional probability is computed as follows:

$$\begin{aligned} P(\text{Variant } k | \text{at most 1}) &= P(\text{at most 1} | \text{Variant } k) * P(\text{Variant } k) / P(\text{at most 1}) \\ &= P(\text{only variant } k) / P(\text{variant } k) * P(\text{Variant } k) / P(\text{at most 1}) \\ &= P(\text{only variant } k) / P(\text{at most 1}) \end{aligned}$$

Where

$$\begin{aligned} P(\text{only variant } k) &= P(k) * \text{Prod}(m \neq v) (1 - P(m)) \\ P(\text{at most 1}) &= P(\text{None}) + \text{Sum}(v \text{ in variants}) P(v) * \text{Prod}(m \neq v) (1 - P(m)) \\ P(\text{None}) &= \text{Prod}(v \text{ in variants}) (1 - P(v)) \end{aligned}$$

Furthermore, the (Variant, Person) pairs are treated independently.

10.5.1 Other parameters

- Who did you get the infection from.
- Omicron is 1.5 more infectious than delta.
- Surveillance:
 - Pull people to be tested at random.
 - Or at symptoms.
 - A mix of the two.
- Define a class for passing extra functions and datasets, for example, testing surveillance.
- Exposed people become infectious after k days.
- [Network](#) changes the can be a function of an ERGM. Apply K steps throughout time.
- Add progress bar.

Chapter 11

Namespace Index

11.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

epiworld::sampler	Functions for sampling viruses	37
sampler	Functions for sampling viruses	40

Chapter 12

Hierarchical Index

12.1 Class Hierarchy

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

AdjList	45
epiworld::AdjList	46
Agent< TSeq >	48
epiworld::Agent< TSeq >	53
Agent< EPI_DEFAULT_TSEQ >	48
AgentsSample< TSeq >	56
epiworld::AgentsSample< TSeq >	58
DataBase< TSeq >	59
epiworld::DataBase< TSeq >	65
Entities< TSeq >	69
epiworld::Entities< TSeq >	70
Entities_const< TSeq >	71
epiworld::Entities_const< TSeq >	72
Entity< TSeq >	73
epiworld::Entity< TSeq >	74
Entity< EPI_DEFAULT_TSEQ >	73
epiworld::Event< TSeq >	76
Event< TSeq >	77
epiworld::GlobalEvent< TSeq >	79
GlobalEvent< TSeq >	81
epiworld::LFMCMC< TData >	82
LFMCMC< TData >	83
epiworld::Model< TSeq >	85
Model< TSeq >	100
ModelMeaslesSchool< TSeq >	142
epiworld::Model< EPI_DEFAULT_TSEQ >	85
ModelDiffNet< TSeq >	119
ModelMeaslesMixing< TSeq >	129
ModelSEIR< TSeq >	149
ModelSEIRCONN< TSeq >	155
ModelSEIRD< TSeq >	161
ModelSEIRDCONN< TSeq >	167
ModelSEIRMixing< TSeq >	174
ModelSEIRMixingQuarantine< TSeq >	186
ModelSIR< TSeq >	195

ModelSIRCONN< TSeq >	201
ModelSIRD< TSeq >	207
ModelSIRDCONN< TSeq >	213
ModelSIRLogit< TSeq >	219
ModelSIRMixing< TSeq >	226
ModelSIS< TSeq >	232
ModelSISD< TSeq >	235
ModelSURV< TSeq >	239
epiworld::epimodels::ModelDiffNet< TSeq >	117
epiworld::epimodels::ModelMeaslesMixing< TSeq >	120
epiworld::epimodels::ModelMeaslesSchool< TSeq >	137
epiworld::epimodels::ModelSEIR< TSeq >	146
epiworld::epimodels::ModelSEIRCONN< TSeq >	152
epiworld::epimodels::ModelSEIRD< TSeq >	158
epiworld::epimodels::ModelSEIRDCONN< TSeq >	164
epiworld::epimodels::ModelSEIRMixing< TSeq >	170
epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >	178
epiworld::epimodels::ModelSIR< TSeq >	194
epiworld::epimodels::ModelSIRCONN< TSeq >	198
epiworld::epimodels::ModelSIRD< TSeq >	204
epiworld::epimodels::ModelSIRDCONN< TSeq >	210
epiworld::epimodels::ModelSIRLogit< TSeq >	215
epiworld::epimodels::ModelSIRMixing< TSeq >	222
epiworld::epimodels::ModelSIS< TSeq >	229
epiworld::epimodels::ModelSISD< TSeq >	233
epiworld::epimodels::ModelSURV< TSeq >	236
epiworld::ModelDiagram	117
ModelDiagram	117
Network< Nettype, Nodetype, Edgetype >	241
epiworld::PersonTools< TSeq >	241
PersonTools< TSeq >	242
epiworld::Progress	242
Progress	242
epiworld::Queue< TSeq >	243
Queue< TSeq >	243
RandGraph	244
epiworld::SAMPLETYPE	245
SAMPLETYPE	245
epiworld::Tool< TSeq >	245
Tool< TSeq >	246
epiworld::ToolFunctions< TSeq >	248
ToolFunctions< TSeq >	249
epiworld::Tools< TSeq >	249
Tools< TSeq >	250
epiworld::Tools_const< TSeq >	251
Tools_const< TSeq >	251
epiworld::UserData< TSeq >	252
UserData< TSeq >	254
epiworld::vecHasher< T >	256
vecHasher< T >	256
epiworld::Virus< TSeq >	257
Virus< TSeq >	259
epiworld::Viruses< TSeq >	261
Viruses< TSeq >	262
epiworld::Viruses_const< TSeq >	263
Viruses_const< TSeq >	263
epiworld::VirusFunctions< TSeq >	264
VirusFunctions< TSeq >	265

Chapter 13

Class Index

13.1 Class List

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

AdjList	45
epiworld::AdjList	46
Agent< TSeq >	
Agent (agents)	48
epiworld::Agent< TSeq >	
Agent (agents)	53
AgentsSample< TSeq >	
Sample of agents	56
epiworld::AgentsSample< TSeq >	
Sample of agents	58
DataBase< TSeq >	
Statistical data about the process	59
epiworld::DataBase< TSeq >	
Statistical data about the process	65
Entities< TSeq >	
Set of Entities (useful for building iterators)	69
epiworld::Entities< TSeq >	
Set of Entities (useful for building iterators)	70
Entities_const< TSeq >	
Set of Entities (const) (useful for iterators)	71
epiworld::Entities_const< TSeq >	
Set of Entities (const) (useful for iterators)	72
Entity< TSeq >	73
epiworld::Entity< TSeq >	74
epiworld::Event< TSeq >	
Event data for update an agent	76
Event< TSeq >	
Event data for update an agent	77
epiworld::GlobalEvent< TSeq >	
Template for a Global Event	79
GlobalEvent< TSeq >	
Template for a Global Event	81
epiworld::LFMCMC< TData >	
Likelihood-Free Markov Chain Monte Carlo	82
LFMCMC< TData >	
Likelihood-Free Markov Chain Monte Carlo	83

epiworld::Model< TSeq >	
Core class of epiworld	85
Model< TSeq >	
Core class of epiworld	100
epiworld::ModelDiagram	117
ModelDiagram	117
epiworld::epimodels::ModelDiffNet< TSeq >	
Template for a Network Diffusion Model	117
ModelDiffNet< TSeq >	
Template for a Network Diffusion Model	119
epiworld::epimodels::ModelMeaslesMixing< TSeq >	
Measles model with population mixing, quarantine, and contact tracing	120
ModelMeaslesMixing< TSeq >	
Measles model with population mixing, quarantine, and contact tracing	129
epiworld::epimodels::ModelMeaslesSchool< TSeq >	
Template for a Measles model with quarantine	137
ModelMeaslesSchool< TSeq >	
Template for a Measles model with quarantine	142
epiworld::epimodels::ModelSEIR< TSeq >	
Template for a Susceptible-Exposed-Infected-Removed (SEIR) model	146
ModelSEIR< TSeq >	
Template for a Susceptible-Exposed-Infected-Removed (SEIR) model	149
epiworld::epimodels::ModelSEIRCONN< TSeq >	152
ModelSEIRCONN< TSeq >	155
epiworld::epimodels::ModelSEIRD< TSeq >	
Template for a Susceptible-Exposed-Infected-Removed-Deceased (SEIRD) model	158
ModelSEIRD< TSeq >	
Template for a Susceptible-Exposed-Infected-Removed-Deceased (SEIRD) model	161
epiworld::epimodels::ModelSEIRDCONN< TSeq >	164
ModelSEIRDCONN< TSeq >	167
epiworld::epimodels::ModelSEIRMixing< TSeq >	170
ModelSEIRMixing< TSeq >	174
epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >	
SEIR model with mixing, quarantine, and contact tracing	178
ModelSEIRMixingQuarantine< TSeq >	
SEIR model with mixing, quarantine, and contact tracing	186
epiworld::epimodels::ModelSIR< TSeq >	
Template for a Susceptible-Infected-Removed (SIR) model	194
ModelSIR< TSeq >	
Template for a Susceptible-Infected-Removed (SIR) model	195
epiworld::epimodels::ModelSIRCONN< TSeq >	198
ModelSIRCONN< TSeq >	201
epiworld::epimodels::ModelSIRD< TSeq >	
Template for a Susceptible-Infected-Removed-Deceased (SIRD) model	204
ModelSIRD< TSeq >	
Template for a Susceptible-Infected-Removed-Deceased (SIRD) model	207
epiworld::epimodels::ModelSIRDCONN< TSeq >	210
ModelSIRDCONN< TSeq >	213
epiworld::epimodels::ModelSIRLogit< TSeq >	
Template for a Susceptible-Infected-Removed (SIR) model	215
ModelSIRLogit< TSeq >	
Template for a Susceptible-Infected-Removed (SIR) model	219
epiworld::epimodels::ModelSIRMixing< TSeq >	222
ModelSIRMixing< TSeq >	226
epiworld::epimodels::ModelSIS< TSeq >	
Template for a Susceptible-Infected-Susceptible (SIS) model	229
ModelSIS< TSeq >	
Template for a Susceptible-Infected-Susceptible (SIS) model	232

epiworld::epimodels::ModelSISD< TSeq >	
Template for a Susceptible-Infected-Susceptible-Deceased (SISD) model	233
ModelSISD< TSeq >	
Template for a Susceptible-Infected-Susceptible-Deceased (SISD) model	235
epiworld::epimodels::ModelSURV< TSeq >	236
ModelSURV< TSeq >	239
Network< Nettype, Nodetype, Edgetype >	241
epiworld::PersonTools< TSeq >	241
PersonTools< TSeq >	242
epiworld::Progress	
A simple progress bar	242
Progress	
A simple progress bar	242
epiworld::Queue< TSeq >	
Controls which agents are verified at each step	243
Queue< TSeq >	
Controls which agents are verified at each step	243
RandGraph	244
epiworld::SAMPLETYPE	245
SAMPLETYPE	245
epiworld::Tool< TSeq >	
Tools for defending the agent against the virus	245
Tool< TSeq >	
Tools for defending the agent against the virus	246
epiworld::ToolFunctions< TSeq >	
Helper class to store the functions avoiding multiple shared_pointers (we have only one for the four of these)	248
ToolFunctions< TSeq >	
Helper class to store the functions avoiding multiple shared_pointers (we have only one for the four of these)	249
epiworld::Tools< TSeq >	
Set of tools (useful for building iterators)	249
Tools< TSeq >	
Set of tools (useful for building iterators)	250
epiworld::Tools_const< TSeq >	
Set of Tools (const) (useful for iterators)	251
Tools_const< TSeq >	
Set of Tools (const) (useful for iterators)	251
epiworld::UserData< TSeq >	
Personalized data by the user	252
UserData< TSeq >	
Personalized data by the user	254
epiworld::vecHasher< T >	
Vector hasher	256
vecHasher< T >	
Vector hasher	256
epiworld::Virus< TSeq >	
Virus	257
Virus< TSeq >	
Virus	259
epiworld::Viruses< TSeq >	
Set of viruses (useful for building iterators)	261
Viruses< TSeq >	
Set of viruses (useful for building iterators)	262
epiworld::Viruses_const< TSeq >	
Set of Viruses (const) (useful for iterators)	263
Viruses_const< TSeq >	
Set of Viruses (const) (useful for iterators)	263

epiworld::VirusFunctions< TSeq >	264
VirusFunctions< TSeq >	265

Chapter 14

File Index

14.1 File List

Here is a list of all documented files with brief descriptions:

epiworld.hpp	??
include/epiworld/ adjlist-bones.hpp	??
include/epiworld/ adjlist-meat.hpp	??
include/epiworld/ agent-bones.hpp	??
include/epiworld/ agent-events-meat.hpp	??
include/epiworld/ agent-meat-state.hpp	??
Sampling functions are getting big, so we keep them in a separate file	267
include/epiworld/ agent-meat-virus-sampling.hpp	??
include/epiworld/ agent-meat.hpp	??
include/epiworld/ agentssample-bones.hpp	??
include/epiworld/ config.hpp	??
include/epiworld/ database-bones.hpp	??
include/epiworld/ database-meat.hpp	??
include/epiworld/ entities-bones.hpp	??
include/epiworld/ entity-bones.hpp	??
include/epiworld/ entity-distribute-meat.hpp	??
include/epiworld/ entity-meat.hpp	??
include/epiworld/ epiworld-macros.hpp	??
include/epiworld/ epiworld.hpp	??
include/epiworld/ globalevent-bones.hpp	??
include/epiworld/ globalevent-meat.hpp	??
include/epiworld/ misc.hpp	??
include/epiworld/ model-bones.hpp	??
include/epiworld/ model-meat-print.hpp	??
include/epiworld/ model-meat.hpp	??
include/epiworld/ modeldiagram-bones.hpp	??
include/epiworld/ modeldiagram-meat.hpp	??
include/epiworld/ network-bones.hpp	??
include/epiworld/ progress.hpp	??
include/epiworld/ queue-bones.hpp	??
include/epiworld/ randgraph.hpp	??
include/epiworld/ random_graph.hpp	??
include/epiworld/ seq_processing.hpp	??
include/epiworld/ tool-bones.hpp	??
include/epiworld/ tool-distribute-meat.hpp	??

include/epiworld/ tool-meat.hpp	??
include/epiworld/ tools-bones.hpp	??
include/epiworld/ userdata-bones.hpp	??
include/epiworld/ userdata-meat.hpp	??
include/epiworld/ virus-bones.hpp	??
include/epiworld/ virus-distribute-meat.hpp	??
include/epiworld/ virus-meat.hpp	??
include/epiworld/ viruses-bones.hpp	??
include/epiworld/math/ distributions.hpp	??
include/epiworld/math/ lfmcmc.hpp	??
include/epiworld/math/lfmcmc/ lfmcmc-bones.hpp	??
include/epiworld/math/lfmcmc/ lfmcmc-meat-print.hpp	??
include/epiworld/math/lfmcmc/ lfmcmc-meat.hpp	??
include/epiworld/models/ diffnet.hpp	??
include/epiworld/models/ globalevents.hpp	??
include/epiworld/models/ init-functions.hpp	??
include/epiworld/models/ measlesmixing.hpp	
Template for a Measles model with population mixing, quarantine, and contact tracing	269
include/epiworld/models/ measlesquarantine.hpp	??
include/epiworld/models/ models.hpp	??
include/epiworld/models/ seir.hpp	??
include/epiworld/models/ seirconnected.hpp	??
include/epiworld/models/ seird.hpp	??
include/epiworld/models/ seirdconnected.hpp	??
include/epiworld/models/ seirmixing.hpp	??
include/epiworld/models/ seirmixingquarantine.hpp	
Template for a Susceptible-Exposed-Infected-Removed (SEIR) model with mixing, quarantine, and contact tracing	270
include/epiworld/models/ sir.hpp	??
include/epiworld/models/ sirconnected.hpp	??
include/epiworld/models/ sird.hpp	??
include/epiworld/models/ sirdconnected.hpp	??
include/epiworld/models/ sirlogit.hpp	??
include/epiworld/models/ sirmixing.hpp	??
include/epiworld/models/ sis.hpp	??
include/epiworld/models/ sisd.hpp	??
include/epiworld/models/ surveillance.hpp	??
tests/ tests.hpp	??

Chapter 15

Namespace Documentation

15.1 epiworld::sampler Namespace Reference

Functions for sampling viruses.

Functions

- `template<typename TSeq = EPI_DEFAULT_TSEQ>`
`std::function< void(Agent< TSeq > *, Model< TSeq > *)> make_update_susceptible (std::vector< epiworld_fast_uint > exclude={})`
Make a function to sample from neighbors.
- `template<typename TSeq = EPI_DEFAULT_TSEQ>`
`std::function< Virus< TSeq > *(Agent< TSeq > *, Model< TSeq > *)> make_sample_virus_neighbors (std::vector< epiworld_fast_uint > exclude={})`
Make a function to sample from neighbors.
- `template<typename TSeq = EPI_DEFAULT_TSEQ>`
`Virus< TSeq > * sample_virus_single (Agent< TSeq > *p, Model< TSeq > *m)`
Sample from neighbors pool of viruses (at most one)

15.1.1 Detailed Description

Functions for sampling viruses.

15.1.2 Function Documentation

15.1.2.1 `make_sample_virus_neighbors()`

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
std::function<Virus<TSeq>*(Agent<TSeq>*,Model<TSeq>*)> epiworld::sampler::make_sample_virus_neighbors (
    std::vector< epiworld_fast_uint > exclude = {} ) [inline]
```

Make a function to sample from neighbors.

This is akin to the function `default_update_susceptible`, with the difference that it will create a function that supports excluding states from the sampling frame. For example, individuals who have acquired a virus can be excluded if in incubation state.

Template Parameters

<i>TSeq</i>	
-------------	--

Parameters

<i>exclude</i>	unsigned vector of states that need to be excluded from the sampling
----------------	--

Returns

Virus<TSeq>* of the selected virus. If none selected (or none available,) returns a nullptr;

15.1.2.2 make_update_susceptible()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
std::function<void (Agent<TSeq>*, Model<TSeq>*)> epiworld::sampler::make_update_susceptible (
    std::vector< epiworld_fast_uint > exclude = {} ) [inline]
```

Make a function to sample from neighbors.

This is akin to the function default_update_susceptible, with the difference that it will create a function that supports excluding states from the sampling frame. For example, individuals who have acquired a virus can be excluded if in incubation state.

Template Parameters

<i>TSeq</i>	
-------------	--

Parameters

<i>exclude</i>	unsigned vector of states that need to be excluded from the sampling
----------------	--

Returns

Virus<TSeq>* of the selected virus. If none selected (or none available,) returns a nullptr;

15.1.2.3 sample_virus_single()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
Virus<TSeq>* epiworld::sampler::sample_virus_single (
    Agent< TSeq > * p,
    Model< TSeq > * m ) [inline]
```

Sample from neighbors pool of viruses (at most one)

This function samples at most one virus from the pool of viruses from its neighbors. If no virus is selected, the function returns a `nullptr`, otherwise it returns a pointer to the selected virus.

This can be used to build a new update function (EPI_NEW_UPDATEFUN.)

Template Parameters

<i>TSeq</i>	
-------------	--

Parameters

<i>p</i>	Pointer to person
<i>m</i>	Pointer to the model

Returns

Virus<TSeq>* of the selected virus. If none selected (or none available,) returns a nullptr;

15.2 sampler Namespace Reference

Functions for sampling viruses.

Functions

- template<typename TSeq = EPI_DEFAULT_TSEQ>
std::function< void([Agent](#)< TSeq > *, [Model](#)< TSeq > *)> [make_update_susceptible](#) (std::vector< epiworld_fast_uint > exclude={})
Make a function to sample from neighbors.
- template<typename TSeq = EPI_DEFAULT_TSEQ>
std::function< [Virus](#)< TSeq > *([Agent](#)< TSeq > *, [Model](#)< TSeq > *)> [make_sample_virus_neighbors](#) (std::vector< epiworld_fast_uint > exclude={})
Make a function to sample from neighbors.
- template<typename TSeq = EPI_DEFAULT_TSEQ>
[Virus](#)< TSeq > * [sample_virus_single](#) ([Agent](#)< TSeq > *p, [Model](#)< TSeq > *m)
Sample from neighbors pool of viruses (at most one)

15.2.1 Detailed Description

Functions for sampling viruses.

15.2.2 Function Documentation

15.2.2.1 make_sample_virus_neighbors()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
std::function<Virus<TSeq>*(Agent<TSeq>*,Model<TSeq>*)> sampler::make_sample_virus_neighbors
(
    std::vector< epiworld_fast_uint > exclude = {} ) [inline]
```

Make a function to sample from neighbors.

This is akin to the function `default_update_susceptible`, with the difference that it will create a function that supports excluding states from the sampling frame. For example, individuals who have acquired a virus can be excluded if in incubation state.

Template Parameters

<i>TSeq</i>	
-------------	--

Parameters

<i>exclude</i>	unsigned vector of states that need to be excluded from the sampling
----------------	--

Returns

Virus<TSeq>* of the selected virus. If none selected (or none available,) returns a nullptr;

15.2.2.2 make_update_susceptible()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
std::function<void (Agent<TSeq>*, Model<TSeq>*)> sampler::make_update_susceptible (
    std::vector< epiworld_fast_uint > exclude = {} ) [inline]
```

Make a function to sample from neighbors.

This is akin to the function `default_update_susceptible`, with the difference that it will create a function that supports excluding states from the sampling frame. For example, individuals who have acquired a virus can be excluded if in incubation state.

Template Parameters

<i>TSeq</i>	
-------------	--

Parameters

<i>exclude</i>	unsigned vector of states that need to be excluded from the sampling
----------------	--

Returns

Virus<TSeq>* of the selected virus. If none selected (or none available,) returns a nullptr;

15.2.2.3 sample_virus_single()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
Virus<TSeq>* sampler::sample_virus_single (
    Agent< TSeq > * p,
    Model< TSeq > * m ) [inline]
```

Sample from neighbors pool of viruses (at most one)

This function samples at most one virus from the pool of viruses from its neighbors. If no virus is selected, the function returns a `nullptr`, otherwise it returns a pointer to the selected virus.

This can be used to build a new update function (`EPI_NEW_UPDATEFUN.`)

Template Parameters

<i>TSeq</i>	
-------------	--

Parameters

<i>p</i>	Pointer to person
<i>m</i>	Pointer to the model

Returns

Virus<TSeq>* of the selected virus. If none selected (or none available,) returns a nullptr;

Chapter 16

Class Documentation

16.1 AdjList Class Reference

Public Member Functions

- [AdjList](#) (const std::vector< int > &source, const std::vector< int > &target, int size, bool directed)
Construct a new Adj List object.
- **AdjList** ([AdjList](#) &&a)
- **AdjList** (const [AdjList](#) &a)
- [AdjList](#) & **operator=** (const [AdjList](#) &a)
- void [read_edgelist](#) (std::string fn, int size, int skip=0, bool directed=true)
Read an edgelist.
- std::map< int, int > **operator()** (epiworld_fast_uint i) const
- void **print** (epiworld_fast_uint limit=20u) const
- size_t [vcount](#) () const
Number of vertices/nodes in the network.
- size_t [ecount](#) () const
Number of edges/arcs/ties in the network.
- std::vector< std::map< int, int > > & **get_dat** ()
- bool [is_directed](#) () const
true if the network is directed.

16.1.1 Constructor & Destructor Documentation

16.1.1.1 AdjList()

```
AdjList::AdjList (
    const std::vector< int > & source,
    const std::vector< int > & target,
    int size,
    bool directed ) [inline]
```

Construct a new Adj List object.

Ids in the network are assume to range from 0 to size - 1.

Parameters

<i>source</i>	Unsigned int vector with the source
<i>target</i>	Unsigned int vector with the target
<i>size</i>	Number of vertices in the network.
<i>directed</i>	Bool true if the network is directed

16.1.2 Member Function Documentation

16.1.2.1 read_edgelist()

```
void AdjList::read_edgelist (
    std::string fn,
    int size,
    int skip = 0,
    bool directed = true ) [inline]
```

Read an edgelist.

Ids in the network are assume to range from 0 to `size - 1`.

Parameters

<i>fn</i>	Path to the file
<i>skip</i>	Number of lines to skip (e.g., 1 if there's a header)
<i>directed</i>	true if the network is directed
<i>size</i>	Number of vertices in the network.

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

- include/epiworld/adjlist-bones.hpp
- include/epiworld/adjlist-meat.hpp

16.2 epiworld::AdjList Class Reference

Public Member Functions

- [AdjList](#) (const std::vector< int > &source, const std::vector< int > &target, int size, bool directed)
Construct a new Adj List object.
- [AdjList](#) ([AdjList](#) &&a)
- [AdjList](#) (const [AdjList](#) &a)
- [AdjList](#) & [operator=](#) (const [AdjList](#) &a)
- void [read_edgelist](#) (std::string fn, int size, int skip=0, bool directed=true)
Read an edgelist.

- `std::map< int, int > operator()` (`epiworld_fast_uint i`) `const`
- `void print` (`epiworld_fast_uint limit=20u`) `const`
- `size_t vcount` () `const`
Number of vertices/nodes in the network.
- `size_t ecount` () `const`
Number of edges/arcs/ties in the network.
- `std::vector< std::map< int, int > > & get_dat` ()
- `bool is_directed` () `const`
true if the network is directed.

16.2.1 Constructor & Destructor Documentation

16.2.1.1 AdjList()

```
AdjList::AdjList (
    const std::vector< int > & source,
    const std::vector< int > & target,
    int size,
    bool directed ) [inline]
```

Construct a new Adj List object.

Ids in the network are assume to range from 0 to `size - 1`.

Parameters

<i>source</i>	Unsigned int vector with the source
<i>target</i>	Unsigned int vector with the target
<i>size</i>	Number of vertices in the network.
<i>directed</i>	Bool true if the network is directed

16.2.2 Member Function Documentation

16.2.2.1 read_edgelist()

```
void AdjList::read_edgelist (
    std::string fn,
    int size,
    int skip = 0,
    bool directed = true ) [inline]
```

Read an edgelist.

Ids in the network are assume to range from 0 to `size - 1`.

Parameters

<i>fn</i>	Path to the file
<i>skip</i>	Number of lines to skip (e.g., 1 if there's a header)
<i>directed</i>	<code>true</code> if the network is directed
<i>size</i>	Number of vertices in the network.

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

- `epiworld.hpp`

16.3 Agent< TSeq > Class Template Reference

[Agent](#) (agents)

```
#include <agent-bones.hpp>
```

Public Member Functions

- **Agent** ([Agent](#)< TSeq > &&p)
- **Agent** (const [Agent](#)< TSeq > &p)
- [Agent](#)< TSeq > & **operator=** (const [Agent](#)< TSeq > &other_agent)
- int **get_id** () const
Id of the individual.
- VirusPtr< TSeq > & **get_virus** ()
- const VirusPtr< TSeq > & **get_virus** () const
- ToolPtr< TSeq > & **get_tool** (int i)
- [Tools](#)< TSeq > **get_tools** ()
- const [Tools_const](#)< TSeq > **get_tools** () const
- size_t **get_n_tools** () const noexcept
- void **mutate_virus** ()
- void **add_neighbor** ([Agent](#)< TSeq > &p, bool check_source=true, bool check_target=true)
- void **swap_neighbors** ([Agent](#)< TSeq > &other, size_t n_this, size_t n_other)
Swaps neighbors between the current agent and agent other
- std::vector< [Agent](#)< TSeq > * > **get_neighbors** ()
- size_t **get_n_neighbors** () const
- void **change_state** ([Model](#)< TSeq > *model, epiworld_fast_uint new_state, epiworld_fast_int queue=0)
- const unsigned int & **get_state** () const
- void **reset** ()
- bool **has_tool** (epiworld_fast_uint t) const
- bool **has_tool** (std::string name) const
- bool **has_tool** (const [Tool](#)< TSeq > &t) const
- bool **has_virus** (epiworld_fast_uint t) const
- bool **has_virus** (std::string name) const
- bool **has_virus** (const [Virus](#)< TSeq > &v) const
- bool **has_entity** (epiworld_fast_uint t) const
- bool **has_entity** (std::string name) const
- void **print** (bool compressed=false) const
- [Entities](#)< TSeq > **get_entities** ()
- const [Entities_const](#)< TSeq > **get_entities** () const

- const [Entity](#)< TSeq > & **get_entity** (size_t i) const
- [Entity](#)< TSeq > & **get_entity** (size_t i)
- size_t **get_n_entities** () const
- bool **operator==** (const [Agent](#)< TSeq > &other) const
- bool **operator!=** (const [Agent](#)< TSeq > &other) const

Add/Remove Virus/Tool

Any of these is ultimately reflected at the end of the iteration.

Parameters

tool	Tool to add
virus	Virus to add
state_new	state after the change
queue	

- void **add_tool** (ToolPtr< TSeq > &tool, [Model](#)< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **add_tool** (const [Tool](#)< TSeq > &tool, [Model](#)< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **set_virus** (VirusPtr< TSeq > &virus, [Model](#)< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **set_virus** (const [Virus](#)< TSeq > &virus, [Model](#)< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **add_entity** ([Entity](#)< TSeq > &entity, [Model](#)< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **rm_tool** (epiworld_fast_uint tool_idx, [Model](#)< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **rm_tool** (ToolPtr< TSeq > &tool, [Model](#)< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **rm_virus** ([Model](#)< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **rm_entity** (epiworld_fast_uint entity_idx, [Model](#)< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **rm_entity** ([Entity](#)< TSeq > &entity, [Model](#)< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **rm_agent_by_virus** ([Model](#)< TSeq > *model)
[Agent](#) removed by virus.

Get the rates (multipliers) for the agent

Parameters

v	A pointer to a virus.
---	-----------------------

Returns

[epiworld_double](#)

- epiworld_double **get_susceptibility_reduction** (VirusPtr< TSeq > v, [Model](#)< TSeq > *model)
- epiworld_double **get_transmission_reduction** (VirusPtr< TSeq > v, [Model](#)< TSeq > *model)
- epiworld_double **get_recovery_enhancer** (VirusPtr< TSeq > v, [Model](#)< TSeq > *model)
- epiworld_double **get_death_reduction** (VirusPtr< TSeq > v, [Model](#)< TSeq > *model)

- double & [operator\(\)](#) (size_t j)
Access the j-th column of the agent.
- double & **operator[]** (size_t j)
- double **operator()** (size_t j) const
- double **operator[]** (size_t j) const

Friends

- class **Model**< TSeq >
- class **Virus**< TSeq >
- class **Tool**< TSeq >
- class **Tools**< TSeq >
- class **Tools_const**< TSeq >
- class **Queue**< TSeq >
- class **Entities**< TSeq >
- class **AgentsSample**< TSeq >
- void **default_add_virus** ([Event](#)< TSeq > &a, [Model](#)< TSeq > *m)
- void **default_add_tool** ([Event](#)< TSeq > &a, [Model](#)< TSeq > *m)
- void **default_add_entity** ([Event](#)< TSeq > &a, [Model](#)< TSeq > *m)
- void **default_rm_virus** ([Event](#)< TSeq > &a, [Model](#)< TSeq > *m)
- void **default_rm_tool** ([Event](#)< TSeq > &a, [Model](#)< TSeq > *m)
- void **default_rm_entity** ([Event](#)< TSeq > &a, [Model](#)< TSeq > *m)
- void **default_change_state** ([Event](#)< TSeq > &a, [Model](#)< TSeq > *m)

16.3.1 Detailed Description

```
template<typename TSeq>
class Agent< TSeq >
```

[Agent](#) (agents)

Template Parameters

<i>TSeq</i>	Sequence type (should match TSeq across the model)
-------------	--

16.3.2 Member Function Documentation

16.3.2.1 operator()

```
template<typename TSeq >
double & Agent< TSeq >::operator() (
    size_t j ) [inline]
```

Access the j-th column of the agent.

If an external array has been specified, then these two functions can be used to access additional agent's features not included in the model.

The `operator[]` method is with no boundary check, whereas the `operator()` method checks boundaries. The former can result in a segfault.

Parameters

<i>j</i>	
----------	--

Returns

double&

16.3.2.2 swap_neighbors()

```
template<typename TSeq >
void Agent< TSeq >::swap_neighbors (
    Agent< TSeq > & other,
    size_t n_this,
    size_t n_other ) [inline]
```

Swaps neighbors between the current agent and agent *other*

Parameters

<i>other</i>	
<i>n_this</i>	
<i>n_other</i>	

16.3.3 Friends And Related Function Documentation**16.3.3.1 default_rm_entity**

```
template<typename TSeq >
void default_rm_entity (
    Event< TSeq > & a,
    Model< TSeq > * m ) [friend]
```

< Last entity of the agent

< Last agent of the entity

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

- include/epiworld/agent-bones.hpp
- include/epiworld/agent-meat.hpp

16.4 epiworld::Agent< TSeq > Class Template Reference

[Agent](#) (agents)

```
#include <epiworld.hpp>
```

Public Member Functions

- **Agent** ([Agent](#)< TSeq > &&p)
- **Agent** (const [Agent](#)< TSeq > &p)
- [Agent](#)< TSeq > & **operator=** (const [Agent](#)< TSeq > &other_agent)
- int **get_id** () const
Id of the individual.
- VirusPtr< TSeq > & **get_virus** ()
- const VirusPtr< TSeq > & **get_virus** () const
- ToolPtr< TSeq > & **get_tool** (int i)
- [Tools](#)< TSeq > **get_tools** ()
- const [Tools_const](#)< TSeq > **get_tools** () const
- size_t **get_n_tools** () const noexcept
- void **mutate_virus** ()
- void **add_neighbor** ([Agent](#)< TSeq > &p, bool check_source=true, bool check_target=true)
- void **swap_neighbors** ([Agent](#)< TSeq > &other, size_t n_this, size_t n_other)
Swaps neighbors between the current agent and agent other
- std::vector< [Agent](#)< TSeq > * > **get_neighbors** ()
- size_t **get_n_neighbors** () const
- void **change_state** ([Model](#)< TSeq > *model, epiworld_fast_uint new_state, epiworld_fast_int queue=0)
- const unsigned int & **get_state** () const
- void **reset** ()
- bool **has_tool** (epiworld_fast_uint t) const
- bool **has_tool** (std::string name) const
- bool **has_tool** (const [Tool](#)< TSeq > &t) const
- bool **has_virus** (epiworld_fast_uint t) const
- bool **has_virus** (std::string name) const
- bool **has_virus** (const [Virus](#)< TSeq > &v) const
- bool **has_entity** (epiworld_fast_uint t) const
- bool **has_entity** (std::string name) const
- void **print** (bool compressed=false) const
- [Entities](#)< TSeq > **get_entities** ()
- const [Entities_const](#)< TSeq > **get_entities** () const
- const [Entity](#)< TSeq > & **get_entity** (size_t i) const
- [Entity](#)< TSeq > & **get_entity** (size_t i)
- size_t **get_n_entities** () const
- bool **operator==** (const [Agent](#)< TSeq > &other) const
- bool **operator!=** (const [Agent](#)< TSeq > &other) const

Add/Remove Virus/Tool

Any of these is ultimately reflected at the end of the iteration.

Parameters

tool	<i>Tool to add</i>
virus	<i>Virus to add</i>
state_new	<i>state after the change</i>
Generated by Doxygen	

- void **add_tool** (ToolPtr< TSeq > &tool, [Model](#)< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **add_tool** (const [Tool](#)< TSeq > &tool, [Model](#)< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **set_virus** (VirusPtr< TSeq > &virus, [Model](#)< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **set_virus** (const [Virus](#)< TSeq > &virus, [Model](#)< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **add_entity** ([Entity](#)< TSeq > &entity, [Model](#)< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **rm_tool** (epiworld_fast_uint tool_idx, [Model](#)< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **rm_tool** (ToolPtr< TSeq > &tool, [Model](#)< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **rm_virus** ([Model](#)< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **rm_entity** (epiworld_fast_uint entity_idx, [Model](#)< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **rm_entity** ([Entity](#)< TSeq > &entity, [Model](#)< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **rm_agent_by_virus** ([Model](#)< TSeq > *model)
[Agent](#) removed by virus.

Get the rates (multipliers) for the agent

Parameters

v	A pointer to a virus.
---	-----------------------

Returns

epiworld_double

- epiworld_double **get_susceptibility_reduction** (VirusPtr< TSeq > v, [Model](#)< TSeq > *model)
- epiworld_double **get_transmission_reduction** (VirusPtr< TSeq > v, [Model](#)< TSeq > *model)
- epiworld_double **get_recovery_enhancer** (VirusPtr< TSeq > v, [Model](#)< TSeq > *model)
- epiworld_double **get_death_reduction** (VirusPtr< TSeq > v, [Model](#)< TSeq > *model)

- double & **operator()** (size_t j)

Access the j-th column of the agent.

- double & **operator[]** (size_t j)
- double **operator()** (size_t j) const
- double **operator[]** (size_t j) const

Friends

- class **Model**< TSeq >
- class **Virus**< TSeq >
- class **Tool**< TSeq >
- class **Tools**< TSeq >
- class **Tools_const**< TSeq >
- class **Queue**< TSeq >
- class **Entities**< TSeq >
- class **AgentsSample**< TSeq >

- void **default_add_virus** ([Event](#)< TSeq > &a, [Model](#)< TSeq > *m)
- void **default_add_tool** ([Event](#)< TSeq > &a, [Model](#)< TSeq > *m)
- void **default_add_entity** ([Event](#)< TSeq > &a, [Model](#)< TSeq > *m)
- void **default_rm_virus** ([Event](#)< TSeq > &a, [Model](#)< TSeq > *m)
- void **default_rm_tool** ([Event](#)< TSeq > &a, [Model](#)< TSeq > *m)
- void **default_rm_entity** ([Event](#)< TSeq > &a, [Model](#)< TSeq > *m)
- void **default_change_state** ([Event](#)< TSeq > &a, [Model](#)< TSeq > *m)

16.4.1 Detailed Description

```
template<typename TSeq>
class epiworld::Agent< TSeq >
```

[Agent](#) (agents)

Template Parameters

<i>TSeq</i>	Sequence type (should match TSeq across the model)
-------------	--

16.4.2 Member Function Documentation

16.4.2.1 operator()()

```
template<typename TSeq >
double & Agent< TSeq >::operator() (
    size_t j ) [inline]
```

Access the j-th column of the agent.

If an external array has been specified, then these two functions can be used to access additional agent's features not included in the model.

The `operator[]` method is with no boundary check, whereas the `operator()` method checks boundaries. The former can result in a segfault.

Parameters

<i>j</i>	
----------	--

Returns

double&

16.4.2.2 swap_neighbors()

```
template<typename TSeq >
void Agent< TSeq >::swap_neighbors (
    Agent< TSeq > & other,
    size_t n_this,
    size_t n_other ) [inline]
```

Swaps neighbors between the current agent and agent `other`

Parameters

<i>other</i>	
<i>n_this</i>	
<i>n_other</i>	

16.4.3 Friends And Related Function Documentation

16.4.3.1 default_rm_entity

```
template<typename TSeq >
void default_rm_entity (
    Event< TSeq > & a,
    Model< TSeq > * m ) [friend]
```

< Last entity of the agent

< Last agent of the entity

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

- epiworld.hpp

16.5 AgentsSample< TSeq > Class Template Reference

Sample of agents.

```
#include <agentssample-bones.hpp>
```


Public Member Functions

- [AgentsSample](#) ()=delete
Default constructor.
- [AgentsSample](#) (const [AgentsSample](#)< TSeq > &a)=delete
Copy constructor.
- [AgentsSample](#) ([AgentsSample](#)< TSeq > &&a)=delete
Move constructor.
- **AgentsSample** ([Model](#)< TSeq > &model_, size_t n, std::vector< size_t > states_={}, bool truncate=false)
- **AgentsSample** ([Model](#)< TSeq > *model, [Entity](#)< TSeq > &entity_, size_t n, std::vector< size_t > states_←_={}, bool truncate=false)
- [AgentsSample](#) ([Model](#)< TSeq > *model, [Agent](#)< TSeq > &agent_, size_t n, std::vector< size_t > states_←_={}, bool truncate=false)
Sample from the agent's entities.
- std::vector< [Agent](#)< TSeq > * >::iterator **begin** ()
- std::vector< [Agent](#)< TSeq > * >::iterator **end** ()
- [Agent](#)< TSeq > * **operator[]** (size_t n)
- [Agent](#)< TSeq > * **operator()** (size_t n)
- size_t **size** () const noexcept

16.5.1 Detailed Description

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
class AgentsSample< TSeq >
```

Sample of agents.

This class allows sampling agents from Entity<TSeq> and Model<TSeq>.

Template Parameters

<i>TSeq</i>	
-------------	--

16.5.2 Constructor & Destructor Documentation

16.5.2.1 AgentsSample()

```
template<typename TSeq >
AgentsSample< TSeq >::AgentsSample (
    Model< TSeq > * model,
    Agent< TSeq > & agent_,
    size_t n,
    std::vector< size_t > states_ = {},
    bool truncate = false ) [inline]
```

Sample from the agent's entities.

For example, how many individuals the agent contacts in a given point in time.

Template Parameters

<i>TSeq</i>	
-------------	--

Parameters

<i>agent</i> ↔	
—	
<i>n</i>	Sample size
<i>truncate</i>	If the agent has fewer than <i>n</i> connections, then <i>truncate</i> = true will automatically reduce the number of possible samples. Otherwise, if false, then it returns an error.

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

- include/epiworld/agentssample-bones.hpp

16.6 epiworld::AgentsSample< TSeq > Class Template Reference

Sample of agents.

```
#include <epiworld.hpp>
```

Public Member Functions

- [AgentsSample](#) ()=delete
Default constructor.
- [AgentsSample](#) (const [AgentsSample](#)< TSeq > &a)=delete
Copy constructor.
- [AgentsSample](#) ([AgentsSample](#)< TSeq > &&a)=delete
Move constructor.
- [AgentsSample](#) ([Model](#)< TSeq > &model_, size_t n, std::vector< size_t > states_={}, bool truncate=false)
- [AgentsSample](#) ([Model](#)< TSeq > *model, [Entity](#)< TSeq > &entity_, size_t n, std::vector< size_t > states_↔_={}, bool truncate=false)
- [AgentsSample](#) ([Model](#)< TSeq > *model, [Agent](#)< TSeq > &agent_, size_t n, std::vector< size_t > states_↔_={}, bool truncate=false)
Sample from the agent's entities.
- std::vector< [Agent](#)< TSeq > * >::iterator **begin** ()
- std::vector< [Agent](#)< TSeq > * >::iterator **end** ()
- [Agent](#)< TSeq > * **operator[]** (size_t n)
- [Agent](#)< TSeq > * **operator()** (size_t n)
- size_t **size** () const noexcept

16.6.1 Detailed Description

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
class epiworld::AgentsSample< TSeq >
```

Sample of agents.

This class allows sampling agents from [Entity](#)<TSeq> and [Model](#)<TSeq>.

Template Parameters

<i>TSeq</i>	
-------------	--

16.6.2 Constructor & Destructor Documentation

16.6.2.1 AgentsSample()

```
template<typename TSeq >
AgentsSample< TSeq >::AgentsSample (
    Model< TSeq > * model,
    Agent< TSeq > & agent_,
    size_t n,
    std::vector< size_t > states_ = {},
    bool truncate = false ) [inline]
```

Sample from the agent's entities.

For example, how many individuals the agent contacts in a given point in time.

Template Parameters

<i>TSeq</i>	
-------------	--

Parameters

<i>agent</i> ↔	
<i>n</i>	Sample size
<i>truncate</i>	If the agent has fewer than <i>n</i> connections, then <i>truncate</i> = true will automatically reduce the number of possible samples. Otherwise, if false, then it returns an error.

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

- epiworld.hpp

16.7 DataBase< TSeq > Class Template Reference

Statistical data about the process.

```
#include <database-bones.hpp>
```

Public Member Functions

- **DataBase** ([Model](#)< TSeq > &m)
- **DataBase** (const [DataBase](#)< TSeq > &db)
- void [record_virus](#) ([Virus](#)< TSeq > &v)
 - Registering a new variant.*
- void [record_tool](#) ([Tool](#)< TSeq > &t)
- void [set_seq_hasher](#) (std::function< std::vector< int >(TSeq)> fun)
- void [reset](#) ()
- [Model](#)< TSeq > * [get_model](#) ()
- void [record](#) ()
- const std::vector< TSeq > & [get_sequence](#) () const
- const std::vector< int > & [get_nexposed](#) () const
- size_t [size](#) () const
- void [write_data](#) (std::string fn_virus_info, std::string fn_virus_hist, std::string fn_tool_info, std::string fn_↵
_tool_hist, std::string fn_total_hist, std::string fn_transmission, std::string fn_transition, std::string fn_↵
reproductive_number, std::string fn_generation_time) const
- void [record_transmission](#) (int i, int j, int virus, int i_expo_date)
- size_t [get_n_viruses](#) () const
 - Get the number of viruses.*
- size_t [get_n_tools](#) () const
 - Get the number of tools.*
- void [set_user_data](#) (std::vector< std::string > names)
- void [add_user_data](#) (std::vector< epiworld_double > x)
- void [add_user_data](#) (epiworld_fast_uint j, epiworld_double x)
- [UserData](#)< TSeq > & [get_user_data](#) ()
- std::vector< epiworld_double > [get_transition_probability](#) (bool print=true, bool normalize=true) const
 - Calculates the transition probabilities.*
- bool [operator==](#) (const [DataBase](#)< TSeq > &other) const
- bool [operator!=](#) (const [DataBase](#)< TSeq > &other) const
- bool [operator==](#) (const [DataBase](#)< std::vector< int >> &other) const
- bool [operator==](#) (const [DataBase](#)< std::vector< int >> &other) const

Get recorded information from the model

Parameters

what	<i>std::string, The state, e.g., 0, 1, 2, ...</i>
------	---

Returns

In [get_today_total](#), the current counts of what.

In [get_today_virus](#), the current counts of what for each virus.

In [get_hist_total](#), the time series of what

In [get_hist_virus](#), the time series of what for each virus.

In [get_hist_total_date](#) and [get_hist_virus_date](#) the corresponding date

- int [get_today_total](#) (const std::string &what) const
- int [get_today_total](#) (const epiworld_fast_uint &what) const
- void [get_today_total](#) (std::vector< std::string > *state=nullptr, std::vector< int > *counts=nullptr) const
- void [get_today_virus](#) (std::vector< std::string > &state, std::vector< int > &id, std::vector< int > &counts) const
- void [get_today_transition_matrix](#) (std::vector< int > &counts) const
- void [get_hist_total](#) (std::vector< int > *date, std::vector< std::string > *state, std::vector< int > *counts) const

- void **get_hist_virus** (std::vector< int > &date, std::vector< int > &id, std::vector< std::string > &state, std::vector< int > &counts) const
 - void **get_hist_tool** (std::vector< int > &date, std::vector< int > &id, std::vector< std::string > &state, std::vector< int > &counts) const
 - void **get_hist_transition_matrix** (std::vector< std::string > &state_from, std::vector< std::string > &state_to, std::vector< int > &date, std::vector< int > &counts, bool skip_zeros) const
-
- void **get_transmissions** (std::vector< int > &date, std::vector< int > &source, std::vector< int > &target, std::vector< int > &virus, std::vector< int > &source_exposure_date) const
Get the transmissions object.
 - void **get_transmissions** (int *date, int *source, int *target, int *virus, int *source_exposure_date) const
-
- MapVec_type< int, int > **get_reproductive_number** () const
Computes the reproductive number of each case.
 - void **get_reproductive_number** (std::string fn) const
-
- void **get_generation_time** (std::vector< int > &agent_id, std::vector< int > &virus_id, std::vector< int > &time, std::vector< int > &gentime) const
Get the generation time.
 - void **get_generation_time** (std::string fn) const
Write the generation time to a file.

Friends

- class **Model**< TSeq >
- void **default_add_virus** (Event< TSeq > &a, Model< TSeq > *m)
- void **default_add_tool** (Event< TSeq > &a, Model< TSeq > *m)
- void **default_rm_virus** (Event< TSeq > &a, Model< TSeq > *m)
- void **default_rm_tool** (Event< TSeq > &a, Model< TSeq > *m)
- void **default_change_state** (Event< TSeq > &a, Model< TSeq > *m)

16.7.1 Detailed Description

```
template<typename TSeq>
class DataBase< TSeq >
```

Statistical data about the process.

Template Parameters

<i>TSeq</i>	
-------------	--

16.7.2 Member Function Documentation

16.7.2.1 `get_generation_time()`

```
template<typename TSeq >
void DataBase< TSeq >::get_generation_time (
    std::vector< int > & agent_id,
    std::vector< int > & virus_id,
    std::vector< int > & time,
    std::vector< int > & gentime ) const [inline]
```

Get the generation time.

Calculates the generating time

Parameters

<i>agent_id, virus_id, time, gentime</i>	vectors where to save the values
--	----------------------------------

The generation time is the time between the infection of the source and the infection of the target.

16.7.2.2 `get_reproductive_number()`

```
template<typename TSeq >
MapVec_type< int, int > DataBase< TSeq >::get_reproductive_number [inline]
```

Computes the reproductive number of each case.

By definition, whereas it computes R0 (basic reproductive number) or Rt/R (the effective reproductive number) will depend on whether the virus is allowed to circulate naïvely or not, respectively.

Parameters

<i>fn</i>	File where to write out the reproductive number.
-----------	--

In the case of `MapVec_type<int, int>`, the key is a vector of 3 integers:

- **Virus** id
- Source id
- Date when the source was infected

16.7.2.3 get_transition_probability()

```
template<typename TSeq >
std::vector< epiworld_double > DataBase< TSeq >::get_transition_probability (
    bool print = true,
    bool normalize = true ) const [inline]
```

Calculates the transition probabilities.

Parameters

<i>print</i>	Print the transition matrix.
<i>normalize</i>	Normalize the transition matrix. Otherwise, it returns raw counts.

The transition matrix is the matrix of the counts of transitions from one state to another. So the ij-th element of the matrix is the number of transitions from state i to state j (when not normalized), or the probability of transitioning from state i to state j (when normalized).

Returns

std::vector< epiworld_double >

16.7.2.4 get_transmissions()

```
template<typename TSeq >
void DataBase< TSeq >::get_transmissions (
    std::vector< int > & date,
    std::vector< int > & source,
    std::vector< int > & target,
    std::vector< int > & virus,
    std::vector< int > & source_exposure_date ) const [inline]
```

Get the transmissions object.

Parameters

<i>date</i>	
<i>source</i>	
<i>target</i>	
<i>virus</i>	
<i>source_exposure_date</i>	

16.7.2.5 operator==() [1/3]

```
bool DataBase< std::vector< int > >::operator== (
    const DataBase< std::vector< int >> & other ) const [inline]
```

< Date of the transmission eve,

< Id of the sour,

< Id of the targ,

< Id of the varia,

< Date when the source acquired the varia,

16.7.2.6 operator==() [2/3]

```
bool DataBase< std::vector< int > >::operator== (
    const DataBase< std::vector< int >> & other ) const [inline]
```

< Date of the transmission eve,

< Id of the sour,

< Id of the targ,

< Id of the varia,

< Date when the source acquired the varia,

16.7.2.7 operator==() [3/3]

```
template<typename TSeq >
bool DataBase< TSeq >::operator== (
    const DataBase< TSeq > & other ) const [inline]
```

< Date of the transmission eve

< Id of the sour

< Id of the targ

< Id of the varia

< Date when the source acquired the varia

16.7.2.8 record_virus()

```
template<typename TSeq >
void DataBase< TSeq >::record_virus (
    Virus< TSeq > & v ) [inline]
```

Registering a new variant.

Parameters

v	Pointer to the new virus. Since viruses are originated in the agent, the numbers simply move around. From the parent virus to the new virus. And the total number of infected does not change.
---	--

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

- include/epiworld/database-bones.hpp
- include/epiworld/database-meat.hpp

16.8 epiworld::DataBase< TSeq > Class Template Reference

Statistical data about the process.

```
#include <epiworld.hpp>
```

Public Member Functions

- **DataBase** ([Model](#)< TSeq > &m)
- **DataBase** (const [DataBase](#)< TSeq > &db)
- void **record_virus** ([Virus](#)< TSeq > &v)
Registering a new variant.
- void **record_tool** ([Tool](#)< TSeq > &t)
- void **set_seq_hasher** (std::function< std::vector< int >(TSeq)> fun)
- void **reset** ()
- [Model](#)< TSeq > * **get_model** ()
- void **record** ()
- const std::vector< TSeq > & **get_sequence** () const
- const std::vector< int > & **get_nexposed** () const
- [size_t](#) **size** () const
- void **write_data** (std::string fn_virus_info, std::string fn_virus_hist, std::string fn_tool_info, std::string fn_tool_hist, std::string fn_total_hist, std::string fn_transmission, std::string fn_transition, std::string fn_reproductive_number, std::string fn_generation_time) const
- void **record_transmission** (int i, int j, int virus, int i_expo_date)
- [size_t](#) **get_n_viruses** () const
Get the number of viruses.
- [size_t](#) **get_n_tools** () const
Get the number of tools.
- void **set_user_data** (std::vector< std::string > names)
- void **add_user_data** (std::vector< [epiworld_double](#) > x)
- void **add_user_data** ([epiworld_fast_uint](#) j, [epiworld_double](#) x)
- [UserData](#)< TSeq > & **get_user_data** ()
- std::vector< [epiworld_double](#) > **get_transition_probability** (bool print=true, bool normalize=true) const
Calculates the transition probabilities.
- bool **operator==** (const [DataBase](#)< TSeq > &other) const
- bool **operator!=** (const [DataBase](#)< TSeq > &other) const

Get recorded information from the model

Parameters

what	<i>std::string, The state, e.g., 0, 1, 2, ...</i>
------	---

Returns

In `get_today_total`, the current counts of what.

In `get_today_virus`, the current counts of what for each virus.

In `get_hist_total`, the time series of what

In `get_hist_virus`, the time series of what for each virus.

In `get_hist_total_date` and `get_hist_virus_date` the corresponding date

- int **get_today_total** (const std::string &what) const
 - int **get_today_total** (const epiworld_fast_uint &what) const
 - void **get_today_total** (std::vector< std::string > *state=nullptr, std::vector< int > *counts=nullptr) const
 - void **get_today_virus** (std::vector< std::string > &state, std::vector< int > &id, std::vector< int > &counts) const
 - void **get_today_transition_matrix** (std::vector< int > &counts) const
 - void **get_hist_total** (std::vector< int > *date, std::vector< std::string > *state, std::vector< int > *counts) const
 - void **get_hist_virus** (std::vector< int > &date, std::vector< int > &id, std::vector< std::string > &state, std::vector< int > &counts) const
 - void **get_hist_tool** (std::vector< int > &date, std::vector< int > &id, std::vector< std::string > &state, std::vector< int > &counts) const
 - void **get_hist_transition_matrix** (std::vector< std::string > &state_from, std::vector< std::string > &state_to, std::vector< int > &date, std::vector< int > &counts, bool skip_zeros) const
-
- void **get_transmissions** (std::vector< int > &date, std::vector< int > &source, std::vector< int > &target, std::vector< int > &virus, std::vector< int > &source_exposure_date) const
Get the transmissions object.
 - void **get_transmissions** (int *date, int *source, int *target, int *virus, int *source_exposure_date) const
-
- MapVec_type< int, int > **get_reproductive_number** () const
Computes the reproductive number of each case.
 - void **get_reproductive_number** (std::string fn) const
-
- void **get_generation_time** (std::vector< int > &agent_id, std::vector< int > &virus_id, std::vector< int > &time, std::vector< int > &gentime) const
Get the generation time.
 - void **get_generation_time** (std::string fn) const
Write the generation time to a file.

Friends

- class **Model**< TSeq >
- void **default_add_virus** ([Event](#)< TSeq > &a, [Model](#)< TSeq > *m)
- void **default_add_tool** ([Event](#)< TSeq > &a, [Model](#)< TSeq > *m)
- void **default_rm_virus** ([Event](#)< TSeq > &a, [Model](#)< TSeq > *m)
- void **default_rm_tool** ([Event](#)< TSeq > &a, [Model](#)< TSeq > *m)
- void **default_change_state** ([Event](#)< TSeq > &a, [Model](#)< TSeq > *m)

16.8.1 Detailed Description

```
template<typename TSeq>
class epiworld::DataBase< TSeq >
```

Statistical data about the process.

Template Parameters

<i>TSeq</i>	
-------------	--

16.8.2 Member Function Documentation

16.8.2.1 get_generation_time()

```
template<typename TSeq >
void DataBase< TSeq >::get_generation_time (
    std::vector< int > & agent_id,
    std::vector< int > & virus_id,
    std::vector< int > & time,
    std::vector< int > & gentime ) const [inline]
```

Get the generation time.

Calculates the generating time

Parameters

<i>agent_id, virus_id, time, gentime</i>	vectors where to save the values
--	----------------------------------

The generation time is the time between the infection of the source and the infection of the target.

16.8.2.2 get_reproductive_number()

```
template<typename TSeq >
MapVec_type< int, int > DataBase< TSeq >::get_reproductive_number [inline]
```

Computes the reproductive number of each case.

By definition, whereas it computes R_0 (basic reproductive number) or R_t/R (the effective reproductive number) will depend on whether the virus is allowed to circulate naïvely or not, respectively.

Parameters

<i>fn</i>	File where to write out the reproductive number.
-----------	--

In the case of `MapVec_type<int, int>`, the key is a vector of 3 integers:

- [Virus](#) id
- Source id
- Date when the source was infected

16.8.2.3 get_transition_probability()

```
template<typename TSeq >
std::vector< epiworld_double > DataBase< TSeq >::get_transition_probability (
    bool print = true,
    bool normalize = true ) const [inline]
```

Calculates the transition probabilities.

Parameters

<i>print</i>	Print the transition matrix.
<i>normalize</i>	Normalize the transition matrix. Otherwise, it returns raw counts.

The transition matrix is the matrix of the counts of transitions from one state to another. So the ij -th element of the matrix is the number of transitions from state i to state j (when not normalized), or the probability of transitioning from state i to state j (when normalized).

Returns

`std::vector< epiworld_double >`

16.8.2.4 get_transmissions()

```
template<typename TSeq >
void DataBase< TSeq >::get_transmissions (
    std::vector< int > & date,
    std::vector< int > & source,
    std::vector< int > & target,
    std::vector< int > & virus,
    std::vector< int > & source_exposure_date ) const [inline]
```

Get the transmissions object.

Parameters

<i>date</i>	
<i>source</i>	
<i>target</i>	
<i>virus</i>	
<i>source_exposure_date</i>	

16.8.2.5 operator==()

```
template<typename TSeq >
bool DataBase< TSeq >::operator== (
    const DataBase< TSeq > & other ) const [inline]
```

< Date of the transmission eve

< Id of the sour

< Id of the targ

< Id of the varia

< Date when the source acquired the varia

16.8.2.6 record_virus()

```
template<typename TSeq >
void DataBase< TSeq >::record_virus (
    Virus< TSeq > & v ) [inline]
```

Registering a new variant.

Parameters

<i>v</i>	Pointer to the new virus. Since viruses are originated in the agent, the numbers simply move around. From the parent virus to the new virus. And the total number of infected does not change.
----------	--

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

- epiworld.hpp

16.9 Entities< TSeq > Class Template Reference

Set of [Entities](#) (useful for building iterators)

#include <entities-bones.hpp>

Public Member Functions

- **Entities** ([Agent](#)< TSeq > &p)
- std::vector< [Entity](#)< TSeq > * >::iterator **begin** ()
- std::vector< [Entity](#)< TSeq > * >::iterator **end** ()
- [Entity](#)< TSeq > & **operator**() (size_t i)
- [Entity](#)< TSeq > & **operator**[] (size_t i)
- size_t **size** () const noexcept
- bool **operator==** (const [Entities](#)< TSeq > &other) const

Friends

- class **Entity**< TSeq >
- class **Agent**< TSeq >

16.9.1 Detailed Description

```
template<typename TSeq>
class Entities< TSeq >
```

Set of [Entities](#) (useful for building iterators)

Template Parameters

<i>TSeq</i>	
-------------	--

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

- include/epiworld/agent-bones.hpp
- include/epiworld/entities-bones.hpp

16.10 epiworld::Entities< TSeq > Class Template Reference

Set of [Entities](#) (useful for building iterators)

```
#include <epiworld.hpp>
```

Public Member Functions

- **Entities** ([Agent](#)< TSeq > &p)
- std::vector< [Entity](#)< TSeq > * >::iterator **begin** ()
- std::vector< [Entity](#)< TSeq > * >::iterator **end** ()
- [Entity](#)< TSeq > & **operator**() (size_t i)
- [Entity](#)< TSeq > & **operator**[] (size_t i)
- size_t **size** () const noexcept
- bool **operator==** (const [Entities](#)< TSeq > &other) const

Friends

- class **Entity**< TSeq >
- class **Agent**< TSeq >

16.10.1 Detailed Description

```
template<typename TSeq>
class epiworld::Entities< TSeq >
```

Set of [Entities](#) (useful for building iterators)

Template Parameters

<i>TSeq</i>	
-------------	--

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

- epiworld.hpp

16.11 Entities_const< TSeq > Class Template Reference

Set of [Entities](#) (const) (useful for iterators)

```
#include <entities-bones.hpp>
```

Public Member Functions

- **Entities_const** (const [Agent](#)< TSeq > &p)
- std::vector< [Entity](#)< TSeq > * >::const_iterator **begin** ()
- std::vector< [Entity](#)< TSeq > * >::const_iterator **end** ()
- const [Entity](#)< TSeq > & **operator()** (size_t i)
- const [Entity](#)< TSeq > & **operator[]** (size_t i)
- size_t **size** () const noexcept
- bool **operator==** (const [Entities_const](#)< TSeq > &other) const

Friends

- class **Virus**< TSeq >
- class **Agent**< TSeq >

16.11.1 Detailed Description

```
template<typename TSeq>
class Entities_const< TSeq >
```

Set of [Entities](#) (const) (useful for iterators)

Template Parameters

<i>TSeq</i>	
-------------	--

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

- include/epiworld/entities-bones.hpp

16.12 epiworld::Entities_const< TSeq > Class Template Reference

Set of [Entities](#) (const) (useful for iterators)

```
#include <epiworld.hpp>
```

Public Member Functions

- **Entities_const** (const [Agent](#)< TSeq > &p)
- std::vector< [Entity](#)< TSeq > * >::const_iterator **begin** ()
- std::vector< [Entity](#)< TSeq > * >::const_iterator **end** ()
- const [Entity](#)< TSeq > & **operator()** (size_t i)
- const [Entity](#)< TSeq > & **operator[]** (size_t i)
- size_t **size** () const noexcept
- bool **operator==** (const [Entities_const](#)< TSeq > &other) const

Friends

- class **Virus**< TSeq >
- class **Agent**< TSeq >

16.12.1 Detailed Description

```
template<typename TSeq>
class epiworld::Entities_const< TSeq >
```

Set of [Entities](#) (const) (useful for iterators)

Template Parameters

<i>TSeq</i>	
-------------	--

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

- epiworld.hpp

16.13 Entity< TSeq > Class Template Reference

Public Member Functions

- [Entity](#) (std::string name, EntityToAgentFun< TSeq > fun=nullptr)
Constructs an [Entity](#) object.
- void **add_agent** ([Agent](#)< TSeq > &p, [Model](#)< TSeq > *model)
- void **add_agent** ([Agent](#)< TSeq > *p, [Model](#)< TSeq > *model)
- void **rm_agent** (size_t idx, [Model](#)< TSeq > *model)
- size_t **size** () const noexcept
- void **set_location** (std::vector< epiworld_double > loc)
- std::vector< epiworld_double > & **get_location** ()
- std::vector< [Agent](#)< TSeq > * >::iterator **begin** ()
- std::vector< [Agent](#)< TSeq > * >::iterator **end** ()
- std::vector< [Agent](#)< TSeq > * >::const_iterator **begin** () const
- std::vector< [Agent](#)< TSeq > * >::const_iterator **end** () const
- size_t **operator[]** (size_t i)
- int **get_id** () const noexcept
- const std::string & **get_name** () const noexcept
- void **set_state** (epiworld_fast_int init, epiworld_fast_int post)
- void **set_queue** (epiworld_fast_int init, epiworld_fast_int post)
- void **get_state** (epiworld_fast_int *init, epiworld_fast_int *post)
- void **get_queue** (epiworld_fast_int *init, epiworld_fast_int *post)
- void **reset** ()
- bool **operator==** (const [Entity](#)< TSeq > &other) const
- bool **operator!=** (const [Entity](#)< TSeq > &other) const

Entity distribution

These functions are used for distributing agents among entities. The idea is to have a flexible way of distributing agents among entities.

- void **distribute** ([Model](#)< TSeq > *model)
- std::vector< size_t > & **get_agents** ()
- void **print** () const
- void **set_distribution** (EntityToAgentFun< TSeq > fun)

Friends

- class [Agent](#)< TSeq >
- class [AgentsSample](#)< TSeq >
- class [Model](#)< TSeq >
- void **default_add_entity** ([Event](#)< TSeq > &a, [Model](#)< TSeq > *m)
- void **default_rm_entity** ([Event](#)< TSeq > &a, [Model](#)< TSeq > *m)

16.13.1 Constructor & Destructor Documentation

16.13.1.1 Entity()

```
template<typename TSeq >
Entity< TSeq >::Entity (
    std::string name,
    EntityToAgentFun< TSeq > fun = nullptr ) [inline]
```

Constructs an [Entity](#) object.

This constructor initializes an [Entity](#) object with the specified parameters.

Parameters

<i>name</i>	The name of the entity.
<i>fun</i>	A function pointer to a function that maps the entity to an agent.

16.13.2 Friends And Related Function Documentation

16.13.2.1 default_rm_entity

```
template<typename TSeq >
void default_rm_entity (
    Event< TSeq > & a,
    Model< TSeq > * m ) [friend]
```

< Last entity of the agent

< Last agent of the entity

< Last entity of the agent

< Last agent of the entity

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

- include/epiworld/agent-bones.hpp
- include/epiworld/entity-bones.hpp
- include/epiworld/entity-meat.hpp

16.14 epiworld::Entity< TSeq > Class Template Reference

Public Member Functions

- [Entity](#) (std::string name, EntityToAgentFun< TSeq > fun=nullptr)
Constructs an [Entity](#) object.
- void **add_agent** ([Agent](#)< TSeq > &p, [Model](#)< TSeq > *model)
- void **add_agent** ([Agent](#)< TSeq > *p, [Model](#)< TSeq > *model)
- void **rm_agent** (size_t idx, [Model](#)< TSeq > *model)
- size_t **size** () const noexcept
- void **set_location** (std::vector< epiworld_double > loc)
- std::vector< epiworld_double > & **get_location** ()
- std::vector< [Agent](#)< TSeq > * >::iterator **begin** ()
- std::vector< [Agent](#)< TSeq > * >::iterator **end** ()
- std::vector< [Agent](#)< TSeq > * >::const_iterator **begin** () const
- std::vector< [Agent](#)< TSeq > * >::const_iterator **end** () const
- size_t **operator[]** (size_t i)
- int **get_id** () const noexcept
- const std::string & **get_name** () const noexcept

- void **set_state** (epiworld_fast_int init, epiworld_fast_int post)
- void **set_queue** (epiworld_fast_int init, epiworld_fast_int post)
- void **get_state** (epiworld_fast_int *init, epiworld_fast_int *post)
- void **get_queue** (epiworld_fast_int *init, epiworld_fast_int *post)
- void **reset** ()
- bool **operator==** (const Entity< TSeq > &other) const
- bool **operator!=** (const Entity< TSeq > &other) const

Entity distribution

These functions are used for distributing agents among entities. The idea is to have a flexible way of distributing agents among entities.

- void **distribute** (Model< TSeq > *model)
- std::vector< size_t > & **get_agents** ()
- void **print** () const
- void **set_distribution** (EntityToAgentFun< TSeq > fun)

Friends

- class **Agent**< TSeq >
- class **AgentsSample**< TSeq >
- class **Model**< TSeq >
- void **default_add_entity** (Event< TSeq > &a, Model< TSeq > *m)
- void **default_rm_entity** (Event< TSeq > &a, Model< TSeq > *m)

16.14.1 Constructor & Destructor Documentation

16.14.1.1 Entity()

```
template<typename TSeq >
epiworld::Entity< TSeq >::Entity (
    std::string name,
    EntityToAgentFun< TSeq > fun = nullptr ) [inline]
```

Constructs an Entity object.

This constructor initializes an Entity object with the specified parameters.

Parameters

<i>name</i>	The name of the entity.
<i>fun</i>	A function pointer to a function that maps the entity to an agent.

16.14.2 Friends And Related Function Documentation

16.14.2.1 default_rm_entity

```
template<typename TSeq >
void default_rm_entity (
    Event< TSeq > & a,
    Model< TSeq > * m ) [friend]
```

< Last entity of the agent

< Last agent of the entity

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

- epiworld.hpp

16.15 epiworld::Event< TSeq > Struct Template Reference

[Event](#) data for update an agent.

```
#include <epiworld.hpp>
```

Public Member Functions

- [Event](#) ([Agent](#)< TSeq > *agent_, VirusPtr< TSeq > &virus_, ToolPtr< TSeq > &tool_, [Entity](#)< TSeq > *entity_, epiworld_fast_int new_state_, epiworld_fast_int queue_, EventFun< TSeq > &call_, int idx_agent_, int idx_object_)↵

Construct a new [Event](#) object.

Public Attributes

- [Agent](#)< TSeq > * **agent**
- VirusPtr< TSeq > **virus**
- ToolPtr< TSeq > **tool**
- [Entity](#)< TSeq > * **entity**
- epiworld_fast_int **new_state**
- epiworld_fast_int **queue**
- EventFun< TSeq > **call**
- int **idx_agent**
- int **idx_object**

16.15.1 Detailed Description

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
struct epiworld::Event< TSeq >
```

[Event](#) data for update an agent.

Template Parameters

<i>TSeq</i>	
-------------	--

16.15.2 Constructor & Destructor Documentation

16.15.2.1 Event()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
epiworld::Event< TSeq >::Event (
    Agent< TSeq > * agent_,
    VirusPtr< TSeq > & virus_,
    ToolPtr< TSeq > & tool_,
    Entity< TSeq > * entity_,
    epiworld_fast_int new_state_,
    epiworld_fast_int queue_,
    EventFun< TSeq > & call_,
    int idx_agent_,
    int idx_object_ ) [inline]
```

Construct a new [Event](#) object.

All the parameters are rather optional.

Parameters

<i>agent_</i>	Agent over who the action will happen
<i>virus_</i>	Virus to add
<i>tool_</i>	Tool to add
<i>virus_idx</i>	Index of virus to be removed (if needed)
<i>tool_idx</i>	Index of tool to be removed (if needed)
<i>new_↔ state_</i>	Next state
<i>queue_</i>	Effect on the queue
<i>call_</i>	The action call (if needed)
<i>idx_↔ agent_</i>	Location of agent in object.
<i>idx_↔ object_</i>	Location of object in agent.

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

- epiworld.hpp

16.16 Event< TSeq > Struct Template Reference

[Event](#) data for update an agent.

```
#include <config.hpp>
```

Collaboration diagram for Event< TSeq >:



Public Member Functions

- [Event](#) ([Agent](#)< TSeq > *agent_, VirusPtr< TSeq > &virus_, ToolPtr< TSeq > &tool_, [Entity](#)< TSeq > *entity_, epiworld_fast_int new_state_, epiworld_fast_int queue_, EventFun< TSeq > &call_, int idx_agent←_, int idx_object_)

Construct a new [Event](#) object.

Public Attributes

- [Agent](#)< TSeq > * **agent**
- VirusPtr< TSeq > **virus**
- ToolPtr< TSeq > **tool**
- [Entity](#)< TSeq > * **entity**
- epiworld_fast_int **new_state**
- epiworld_fast_int **queue**
- EventFun< TSeq > **call**
- int **idx_agent**
- int **idx_object**

16.16.1 Detailed Description

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
struct Event< TSeq >
```

[Event](#) data for update an agent.

Template Parameters

<i>TSeq</i>	
-------------	--

16.16.2 Constructor & Destructor Documentation

16.16.2.1 Event()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
Event< TSeq >::Event (
    Agent< TSeq > * agent_,
    VirusPtr< TSeq > & virus_,
    ToolPtr< TSeq > & tool_,
    Entity< TSeq > * entity_,
    epiworld_fast_int new_state_,
    epiworld_fast_int queue_,
    EventFun< TSeq > & call_,
    int idx_agent_,
    int idx_object_ ) [inline]
```

Construct a new [Event](#) object.

All the parameters are rather optional.

Parameters

<i>agent_</i>	Agent over who the action will happen
<i>virus_</i>	Virus to add
<i>tool_</i>	Tool to add
<i>virus_idx</i>	Index of virus to be removed (if needed)
<i>tool_idx</i>	Index of tool to be removed (if needed)
<i>new_↔ state_</i>	Next state
<i>queue_</i>	Effect on the queue
<i>call_</i>	The action call (if needed)
<i>idx_↔ agent_</i>	Location of agent in object.
<i>idx_↔ object_</i>	Location of object in agent.

The documentation for this struct was generated from the following files:

- include/epiworld/agent-bones.hpp
- include/epiworld/config.hpp

16.17 epiworld::GlobalEvent< TSeq > Class Template Reference

Template for a Global [Event](#).

```
#include <epiworld.hpp>
```

Public Member Functions

- [GlobalEvent](#) (GlobalFun< TSeq > fun, std::string name, int day=-99)
Construct a new Global [Event](#) object.
- void **operator()** ([Model](#)< TSeq > *m, int day)
- void **set_name** (std::string name)
- std::string **get_name** () const
- void **set_day** (int day)
- int **get_day** () const
- void **print** () const
- bool **operator==** (const [GlobalEvent](#)< TSeq > &other) const
- bool **operator!=** (const [GlobalEvent](#)< TSeq > &other) const

16.17.1 Detailed Description

```
template<typename TSeq>
class epiworld::GlobalEvent< TSeq >
```

Template for a Global [Event](#).

Global events are functions that [Model](#)<TSeq> executes at the end of a day.

16.17.2 Constructor & Destructor Documentation

16.17.2.1 GlobalEvent()

```
template<typename TSeq >
GlobalEvent< TSeq >::GlobalEvent (
    GlobalFun< TSeq > fun,
    std::string name,
    int day = -99 ) [inline]
```

Construct a new Global [Event](#) object.

Parameters

<i>fun</i>	A function that takes a Model <TSeq> * as argument and returns void.
<i>name</i>	A descriptive name for the action.
<i>day</i>	The day when the action will be executed. If negative, it will be executed every day.

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

- epiworld.hpp

16.18 GlobalEvent< TSeq > Class Template Reference

Template for a Global [Event](#).

```
#include <globalevent-bones.hpp>
```

Public Member Functions

- [GlobalEvent](#) (GlobalFun< TSeq > fun, std::string name, int day=-99)
Construct a new Global [Event](#) object.
- void **operator()** (Model< TSeq > *m, int day)
- void **set_name** (std::string name)
- std::string **get_name** () const
- void **set_day** (int day)
- int **get_day** () const
- void **print** () const
- bool **operator==** (const [GlobalEvent](#)< TSeq > &other) const
- bool **operator!=** (const [GlobalEvent](#)< TSeq > &other) const

16.18.1 Detailed Description

```
template<typename TSeq>
class GlobalEvent< TSeq >
```

Template for a Global [Event](#).

Global events are functions that Model<TSeq> executes at the end of a day.

16.18.2 Constructor & Destructor Documentation

16.18.2.1 GlobalEvent()

```
template<typename TSeq >
GlobalEvent< TSeq >::GlobalEvent (
    GlobalFun< TSeq > fun,
    std::string name,
    int day = -99 ) [inline]
```

Construct a new Global [Event](#) object.

Parameters

<i>fun</i>	A function that takes a Model<TSeq> * as argument and returns void.
<i>name</i>	A descriptive name for the action.
<i>day</i>	The day when the action will be executed. If negative, it will be executed every day.

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

- include/epiworld/globalevent-bones.hpp
- include/epiworld/globalevent-meat.hpp

16.19 epiworld::LFMCMC< TData > Class Template Reference

Likelihood-Free Markov Chain Monte Carlo.

```
#include <epiworld.hpp>
```

Public Member Functions

- void **run** (std::vector< epiworld_double > params_init_, size_t n_samples_, epiworld_double epsilon_, int seed=-1)
- **LFMCMC** (const TData &observed_data_)
- void **set_observed_data** (const TData &observed_data_)
- void **set_proposal_fun** (LFMCMCProposalFun< TData > fun)
- void **set_simulation_fun** (LFMCMCSimFun< TData > fun)
- void **set_summary_fun** (LFMCMCSummaryFun< TData > fun)
- void **set_kernel_fun** (LFMCMCKernelFun< TData > fun)
- void **set_params_names** (std::vector< std::string > names)
- void **set_stats_names** (std::vector< std::string > names)
- size_t **get_n_samples** () const
- size_t **get_n_stats** () const
- size_t **get_n_params** () const
- epiworld_double **get_epsilon** () const
- const std::vector< epiworld_double > & **get_initial_params** () const
- const std::vector< epiworld_double > & **get_current_proposed_params** () const
- const std::vector< epiworld_double > & **get_current_accepted_params** () const
- const std::vector< epiworld_double > & **get_current_proposed_stats** () const
- const std::vector< epiworld_double > & **get_current_accepted_stats** () const
- const std::vector< epiworld_double > & **get_observed_stats** () const
- const std::vector< epiworld_double > & **get_all_sample_params** () const
- const std::vector< epiworld_double > & **get_all_sample_stats** () const
- const std::vector< bool > & **get_all_sample_acceptance** () const
- const std::vector< epiworld_double > & **get_all_sample_drawn_prob** () const
- const std::vector< epiworld_double > & **get_all_sample_kernel_scores** () const
- const std::vector< epiworld_double > & **get_all_accepted_params** () const
- const std::vector< epiworld_double > & **get_all_accepted_stats** () const
- const std::vector< epiworld_double > & **get_all_accepted_kernel_scores** () const
- std::vector< TData > * **get_simulated_data** () const
- std::vector< epiworld_double > **get_mean_params** ()
- std::vector< epiworld_double > **get_mean_stats** ()
- **LFMCMC**< TData > & **verbose_off** ()
- **LFMCMC**< TData > & **verbose_on** ()
- void **print** (size_t burnin=0u) const

Random number generation

Parameters

eng	
-----	--

- void **set_rand_engine** (std::shared_ptr< std::mt19937 > &eng)
- std::shared_ptr< std::mt19937 > & **get_rand_engine** ()
- void **seed** (epiworld_fast_uint s)
- void **set_rand_gamma** (epiworld_double alpha, epiworld_double beta)
- epiworld_double **runif** ()
- epiworld_double **rnorm** ()
- epiworld_double **rgamma** ()
- epiworld_double **runif** (epiworld_double lb, epiworld_double ub)
- epiworld_double **rnorm** (epiworld_double mean, epiworld_double sd)
- epiworld_double **rgamma** (epiworld_double alpha, epiworld_double beta)

16.19.1 Detailed Description

```
template<typename TData>
class epiworld::LFMCMC< TData >
```

Likelihood-Free Markov Chain Monte Carlo.

Template Parameters

<i>TData</i>	Type of data that is generated
--------------	--------------------------------

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

- epiworld.hpp

16.20 LFMCMC< TData > Class Template Reference

Likelihood-Free Markov Chain Monte Carlo.

```
#include <lfmcmc-bones.hpp>
```

Public Member Functions

- void **run** (std::vector< epiworld_double > params_init_, size_t n_samples_, epiworld_double epsilon_, int seed=-1)
- **LFMCMC** (const TData &observed_data_)
- void **set_observed_data** (const TData &observed_data_)
- void **set_proposal_fun** (LFMCMCProposalFun< TData > fun)
- void **set_simulation_fun** (LFMCMCSimFun< TData > fun)
- void **set_summary_fun** (LFMCMCSummaryFun< TData > fun)
- void **set_kernel_fun** (LFMCMCKernelFun< TData > fun)
- void **set_params_names** (std::vector< std::string > names)
- void **set_stats_names** (std::vector< std::string > names)
- size_t **get_n_samples** () const

- `size_t` **get_n_stats** () const
- `size_t` **get_n_params** () const
- `epiworld_double` **get_epsilon** () const
- `const std::vector< epiworld_double >` & **get_initial_params** () const
- `const std::vector< epiworld_double >` & **get_current_proposed_params** () const
- `const std::vector< epiworld_double >` & **get_current_accepted_params** () const
- `const std::vector< epiworld_double >` & **get_current_proposed_stats** () const
- `const std::vector< epiworld_double >` & **get_current_accepted_stats** () const
- `const std::vector< epiworld_double >` & **get_observed_stats** () const
- `const std::vector< epiworld_double >` & **get_all_sample_params** () const
- `const std::vector< epiworld_double >` & **get_all_sample_stats** () const
- `const std::vector< bool >` & **get_all_sample_acceptance** () const
- `const std::vector< epiworld_double >` & **get_all_sample_drawn_prob** () const
- `const std::vector< epiworld_double >` & **get_all_sample_kernel_scores** () const
- `const std::vector< epiworld_double >` & **get_all_accepted_params** () const
- `const std::vector< epiworld_double >` & **get_all_accepted_stats** () const
- `const std::vector< epiworld_double >` & **get_all_accepted_kernel_scores** () const
- `std::vector< TData >` * **get_simulated_data** () const
- `std::vector< epiworld_double >` **get_mean_params** ()
- `std::vector< epiworld_double >` **get_mean_stats** ()
- `LFMCMC< TData >` & **verbose_off** ()
- `LFMCMC< TData >` & **verbose_on** ()
- `void` **print** (size_t burnin=0u) const

Random number generation

Parameters

eng	
-----	--

- `void` **set_rand_engine** (std::shared_ptr< std::mt19937 > &eng)
- `std::shared_ptr< std::mt19937 >` & **get_rand_engine** ()
- `void` **seed** (epiworld_fast_uint s)
- `void` **set_rand_gamma** (epiworld_double alpha, epiworld_double beta)
- `epiworld_double` **runif** ()
- `epiworld_double` **rnorm** ()
- `epiworld_double` **rgamma** ()
- `epiworld_double` **runif** (epiworld_double lb, epiworld_double ub)
- `epiworld_double` **rnorm** (epiworld_double mean, epiworld_double sd)
- `epiworld_double` **rgamma** (epiworld_double alpha, epiworld_double beta)

16.20.1 Detailed Description

```
template<typename TData>
class LFMCMC< TData >
```

Likelihood-Free Markov Chain Monte Carlo.

Template Parameters

<i>TData</i>	Type of data that is generated
--------------	--------------------------------

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

- include/epiworld/math/lfmcmc/lfmcmc-bones.hpp
- include/epiworld/math/lfmcmc/lfmcmc-meat-print.hpp
- include/epiworld/math/lfmcmc/lfmcmc-meat.hpp

16.21 epiworld::Model< TSeq > Class Template Reference

Core class of epiworld.

```
#include <epiworld.hpp>
```

Collaboration diagram for epiworld::Model< TSeq >:



Public Member Functions

- [DataBase< TSeq >](#) & **get_db** ()
- const [DataBase< TSeq >](#) & **get_db** () const
- epiworld_double & **operator()** (std::string pname)
- size_t **size** () const
- void **load_agents_entities_ties** (std::string fn, int skip)
Associate agents-entities from a file.
- void **load_agents_entities_ties** (const std::vector< int > &agents_ids, const std::vector< int > &entities_ids)
Associate agents-entities from data.
- void **load_agents_entities_ties** (const int *agents_id, const int *entities_id, size_t n)
- size_t **get_n_viruses** () const
Number of viruses in the model.
- size_t **get_n_tools** () const
Number of tools in the model.
- epiworld_fast_uint **get_ndays** () const
- epiworld_fast_uint **get_n_replicates** () const
- size_t **get_n_entities** () const
- void **set_ndays** (epiworld_fast_uint ndays)
- bool **get_verbose** () const
- [Model< TSeq >](#) & **verbose_off** ()
- [Model< TSeq >](#) & **verbose_on** ()

- int **today** () const
The current time of the model.
- void **write_data** (std::string fn_virus_info, std::string fn_virus_hist, std::string fn_tool_info, std::string fn_tool_hist, std::string fn_total_hist, std::string fn_transmission, std::string fn_transition, std::string fn_reproductive_number, std::string fn_generation_time) const
Wrapper of DataBase::write_data
- std::map< std::string, epiworld_double > & **params** ()
- virtual void **reset** ()
Reset the model.
- const **Model**< TSeq > & **print** (bool lite=false) const
- void **get_elapsed** (std::string unit="auto", epiworld_double *last_elapsed=nullptr, epiworld_double *total_elapsed=nullptr, std::string *unit_abbr=nullptr, bool print=true) const
- void **add_globalevent** (std::function< void(**Model**< TSeq > *)> fun, std::string name="A global action", int date=-99)
Set a global action.
- void **add_globalevent** (**GlobalEvent**< TSeq > action)
- **GlobalEvent**< TSeq > & **get_globalevent** (std::string name)
Retrieve a global action by name.
- **GlobalEvent**< TSeq > & **get_globalevent** (size_t i)
Retrieve a global action by index.
- void **rm_globalevent** (std::string name)
Remove a global action by name.
- void **rm_globalevent** (size_t i)
Remove a global action by index.
- void **run_globalevents** ()
- void **clear_state_set** ()
- const std::vector< VirusPtr< TSeq > > & **get_viruses** () const
- const std::vector< ToolPtr< TSeq > > & **get_tools** () const
- **Virus**< TSeq > & **get_virus** (size_t id)
- **Tool**< TSeq > & **get_tool** (size_t id)
- void **set_agents_data** (double *data_, size_t ncols_)
Set the agents data object.
- double * **get_agents_data** ()
- size_t **get_agents_data_ncols** () const
- void **set_name** (std::string name)
Set the name object.
- std::string **get_name** () const
- bool **operator==** (const **Model**< TSeq > &other) const
- bool **operator!=** (const **Model**< TSeq > &other) const
- void **events_run** ()
Executes the stored action.
- void **draw** (DiagramType diagram_type=DiagramType::Mermaid, const std::string &fn_output="", bool self=false)
Draws a mermaid diagram of the model.

Set the backup object

backup can be used to restore the entire object after a run. This can be useful if the user wishes to have individuals start with the same network from the beginning.

- void **set_backup** ()

Random number generation

Parameters

eng	<i>Random number generator</i>
s	<i>Seed</i>

- void **set_rand_engine** (std::shared_ptr< std::mt19937 > &eng)
- std::shared_ptr< std::mt19937 > & **get_rand_engine** ()
- void **seed** (size_t s)
- void **set_rand_norm** (epiworld_double mean, epiworld_double sd)
- void **set_rand_unif** (epiworld_double a, epiworld_double b)
- void **set_rand_exp** (epiworld_double lambda)
- void **set_rand_gamma** (epiworld_double alpha, epiworld_double beta)
- void **set_rand_lognormal** (epiworld_double mean, epiworld_double shape)
- void **set_rand_binom** (int n, epiworld_double p)
- void **set_rand_nbinom** (int n, epiworld_double p)
- void **set_rand_geom** (epiworld_double p)
- void **set_rand_pois** (epiworld_double lambda)
- epiworld_double **runif** ()
- epiworld_double **runif** (epiworld_double a, epiworld_double b)
- epiworld_double **rnorm** ()
- epiworld_double **rnorm** (epiworld_double mean, epiworld_double sd)
- epiworld_double **rgamma** ()
- epiworld_double **rgamma** (epiworld_double alpha, epiworld_double beta)
- epiworld_double **rexp** ()
- epiworld_double **rexp** (epiworld_double lambda)
- epiworld_double **rlognormal** ()
- epiworld_double **rlognormal** (epiworld_double mean, epiworld_double shape)
- int **rbinom** ()
- int **rbinom** (int n, epiworld_double p)
- int **rnbinom** ()
- int **rnbinom** (int n, epiworld_double p)
- int **rgeom** ()
- int **rgeom** (epiworld_double p)
- int **rpois** ()
- int **rpois** (epiworld_double lambda)

Add Virus/Tool to the model

This is done before the model has been initialized.

Parameters

v	<i>Virus to be added</i>
t	<i>Tool to be added</i>
preval	<i>Initial prevalence (initial state.) It can be specified as a proportion (between zero and one,) or an integer indicating number of individuals.</i>

- void **add_virus** ([Virus](#)< TSeq > &v)
- void **add_tool** ([Tool](#)< TSeq > &t)
- void **add_entity** ([Entity](#)< TSeq > e)
- void **rm_virus** (size_t virus_pos)
- void **rm_tool** (size_t tool_pos)
- void **rm_entity** (size_t entity_id)

Accessing population of the model

Parameters

fn	<i>std::string Filename of the edgelist file.</i>
skip	<i>int Number of lines to skip in fn.</i>
directed	<i>bool Whether the graph is directed or not.</i>
size	<i>Size of the network.</i>
al	<i>AdjList to read into the model.</i>

- void **agents_from_adjlist** (std::string fn, int size, int skip=0, bool directed=false)
- void **agents_from_edgelist** (const std::vector< int > &source, const std::vector< int > &target, int size, bool directed)
- void **agents_from_adjlist** ([AdjList](#) al)
- bool **is_directed** () const
- std::vector< [Agent](#)< TSeq > > &**get_agents** ()
Returns a reference to the vector of agents.
- [Agent](#)< TSeq > &**get_agent** (size_t i)
- std::vector< epiworld_fast_uint > **get_agents_states** () const
Returns a vector with the states of the agents.
- std::vector< [Viruses](#)_const< TSeq > > **get_agents_viruses** () const
Returns a const vector with the viruses of the agents.
- std::vector< [Viruses](#)< TSeq > > **get_agents_viruses** ()
Returns a vector with the viruses of the agents.
- std::vector< [Entity](#)< TSeq > > &**get_entities** ()
- [Entity](#)< TSeq > &**get_entity** (size_t entity_id, int *entity_pos=nullptr)
- [Model](#)< TSeq > &**agents_smallworld** (epiworld_fast_uint n=1000, epiworld_fast_uint k=5, bool d=false, epiworld_double p=.01)
- void **agents_empty_graph** (epiworld_fast_uint n=1000)

Functions to run the model*Parameters*

seed	<i>Seed to be used for Pseudo-RNG.</i>
ndays	<i>Number of days (steps) of the simulation.</i>
fun	<i>In the case of <code>run_multiple</code>, a function that is called after each experiment.</i>

- void **update_state** ()
- void **mutate_virus** ()
- void **next** ()
- virtual [Model](#)< TSeq > &**run** (epiworld_fast_uint ndays, int seed=-1)
Runs the simulation (after initialization)
- void **run_multiple** (epiworld_fast_uint ndays, epiworld_fast_uint nexperiments, int seed_=-1, std::function< void(size_t, [Model](#)< TSeq > *)> fun=make_save_run< TSeq >(), bool **reset**=true, bool verbose=true, int nthreads=1)

Rewire the network preserving the degree sequence.

This implementation assumes an undirected network, thus if $\{(i,j), (k,l)\} \rightarrow \{(i,l), (k,j)\}$, the reciprocal is also true, i.e., $\{(j,i), (l,k)\} \rightarrow \{(j,k), (l,i)\}$.

Parameters

proportion	<i>Proportion of ties to be rewired.</i>
------------	--

Returns

A rewired version of the network.

- void **set_rewire_fun** (std::function< void(std::vector< [Agent](#)< TSeq >> *, [Model](#)< TSeq > *, epiworld_double)> fun)
- void **set_rewire_prop** (epiworld_double prop)
- epiworld_double **get_rewire_prop** () const
- void **rewire** ()

Export the network data in edgelist form**Parameters**

fn	<i>std::string. File name.</i>
source	<i>Integer vector</i>
target	<i>Integer vector</i>

When passing the source and target, the function will write the edgelist on those.

- void **write_edgelist** (std::string fn) const
- void **write_edgelist** (std::vector< int > &source, std::vector< int > &target) const

Manage state (states) in the model

The functions `get_state` return the current values for the states included in the model.

Parameters

lab	<i>std::string Name of the state.</i>
-----	---------------------------------------

Returns

add_state returns nothing.*

get_state_ returns a vector of pairs with the states and their labels.*

- void **add_state** (std::string lab, UpdateFun< TSeq > fun=nullptr)
- const std::vector< std::string > & **get_states** () const
- size_t **get_n_states** () const
- const std::vector< UpdateFun< TSeq > > & **get_state_fun** () const
- void **print_state_codes** () const

Initial states

These functions are called before the simulation starts.

Parameters

proportions↔	<i>Vector of proportions for each state.</i>
—	
queue_	<i>Vector of queue for each state.</i>

- virtual [Model](#)< TSeq > & **initial_states** (std::vector< double >, std::vector< int >)

Setting and accessing parameters from the model

Tools can incorporate parameters included in the model. Internally, parameters in the tool are stored as pointers to an `std::map<>` of parameters in the model. Using the `epiworld_fast_uint` method directly fetches the parameters in the order these were added to the tool. Accessing parameters via the `std::string` method involves searching the parameter directly in the `std::map<>` member of the model (so it is not recommended.)

The `par()` function members are aliases for `get_param()`.

In the case of the function `read_params`, users can pass a file listing parameters to be included in the model. Each line in the file should have the following structure:

```
[name of parameter 1]: [value in double]
[name of parameter 2]: [value in double]
...
```

The only condition for parameter names is that these do not include a colon.

Parameters

initial_val	
pname	Name of the parameter to add or to fetch
fn	Path to the file containing parameters

Returns

The current value of the parameter in the model.

- `epiworld_double add_param` (`epiworld_double initial_val`, `std::string pname`, `bool overwrite=false`)
- `Model< TSeq > & read_params` (`std::string fn`, `bool overwrite=false`)
- `epiworld_double get_param` (`epiworld_fast_uint k`)
- `epiworld_double get_param` (`std::string pname`)
- `void set_param` (`std::string pname`, `epiworld_double val`)
- `epiworld_double par` (`std::string pname`) `const`

Set the user data object

Parameters

names	string vector with the names of the variables.
-------	--

- `void set_user_data` (`std::vector< std::string > names`)
[@
- `void add_user_data` (`epiworld_fast_uint j`, `epiworld_double x`)
- `void add_user_data` (`std::vector< epiworld_double > x`)
- `UserData< TSeq > & get_user_data` ()

Queuing system

When queuing is on, the model will keep track of which agents are either in risk of exposure or exposed. This then is used at each step to act only on the aforementioned agents.

- `void queuing_on` ()
Activates the queuing system (default.)
- `Model< TSeq > & queuing_off` ()
Deactivates the queuing system.
- `bool is_queuing_on` () `const`
Query if the queuing system is on.
- `Queue< TSeq > & get_queue` ()
Retrieve the `Queue` object.

Get the susceptibility reduction object

Parameters

v	
---	--

Returns

epiworld_double

- void **set_susceptibility_reduction_mixer** (MixerFun< TSeq > fun)
- void **set_transmission_reduction_mixer** (MixerFun< TSeq > fun)
- void **set_recovery_enhancer_mixer** (MixerFun< TSeq > fun)
- void **set_death_reduction_mixer** (MixerFun< TSeq > fun)

Protected Member Functions

- void **dist_tools** ()
- void **dist_virus** ()
- void **dist_entities** ()
- void **chrono_start** ()
- void **chrono_end** ()
- void **events_add** (Agent< TSeq > *agent_, VirusPtr< TSeq > virus_, ToolPtr< TSeq > tool_, Entity< TSeq > *entity_, epiworld_fast_int new_state_, epiworld_fast_int queue_, EventFun< TSeq > call_, int idx_↵ agent_, int idx_object_)

Construct a new [Event](#) object.

Protected Attributes

- std::string **name** = ""
Name of the model.
- DataBase< TSeq > **db** = DataBase<TSeq>(*this)
- std::vector< Agent< TSeq > > **population** = {}
- bool **using_backup** = true
- std::vector< Agent< TSeq > > **population_backup** = {}
- bool **directed** = false
- std::vector< VirusPtr< TSeq > > **viruses** = {}
- std::vector< ToolPtr< TSeq > > **tools** = {}
- std::vector< Entity< TSeq > > **entities** = {}
- std::vector< Entity< TSeq > > **entities_backup** = {}
- std::shared_ptr< std::mt19937 > **engine** = std::make_shared< std::mt19937 >()
- std::uniform_real_distribution **runifd**
- std::normal_distribution **rnormd**
- std::gamma_distribution **rgammad**
- std::lognormal_distribution **rlognormald**
- std::exponential_distribution **rexp**
- std::binomial_distribution **rbinomd**
- std::negative_binomial_distribution **rnbinomd**
- std::geometric_distribution **rgeomd**
- std::poisson_distribution **rpoissd**
- std::function< void(std::vector< Agent< TSeq > > *, Model< TSeq > *, epiworld_double)> **rewire_fun**
- epiworld_double **rewire_prop** = 0.0
- std::map< std::string, epiworld_double > **parameters**
- epiworld_fast_uint **ndays** = 0
- [Progress](#) **pb**

- `std::vector< UpdateFun< TSeq > > state_fun = {}`
Functions to update states.
- `std::vector< std::string > states_labels = {}`
Labels of the states.
- `std::function< void(Model< TSeq > *)> initial_states_fun`
- `epiworld_fast_uint nstates = 0u`
- `bool verbose = true`
- `int current_date = 0`
- `std::chrono::time_point< std::chrono::steady_clock > time_start`
- `std::chrono::time_point< std::chrono::steady_clock > time_end`
- `std::chrono::duration< epiworld_double, std::micro > time_elapsed`
- `epiworld_fast_uint n_replicates = 0u`
- `std::vector< GlobalEvent< TSeq > > globalevents`
- `Queue< TSeq > queue`
- `bool use_queueing = true`
- `std::vector< Event< TSeq > > events = {}`
Variables used to keep track of the events to be made regarding viruses.
- `epiworld_fast_uint nactions = 0u`

Auxiliary variables for AgentsSample<TSeq> iterators

These variables+objects are used by the AgentsSample<TSeq> class for building efficient iterators over agents. The idea is to reduce the memory allocation, so only during the first call of AgentsSample<TSeq>::AgentsSample(Model<TSeq>) these vectors are allocated.

- `std::vector< Agent< TSeq > * > sampled_population`
- `size_t sampled_population_n = 0u`
- `std::vector< size_t > population_left`
- `size_t population_left_n = 0u`

Agents features

Optionally, a model can include an external data source pointing to agents information. The data can then be access through the Agent::operator() method.

- `double * agents_data = nullptr`
- `size_t agents_data_ncols = 0u`

Friends

- `class Agent< TSeq >`
- `class AgentsSample< TSeq >`
- `class DataBase< TSeq >`
- `class Queue< TSeq >`

Tool Mixers

These functions combine the effects tools have to deliver a single effect. For example, wearing a mask, been vaccinated, and the immune system combine together to jointly reduce the susceptibility for a given virus.

- `MixerFun< TSeq > susceptibility_reduction_mixer = susceptibility_reduction_mixer_default<TSeq>`
- `MixerFun< TSeq > transmission_reduction_mixer = transmission_reduction_mixer_default<TSeq>`
- `MixerFun< TSeq > recovery_enhancer_mixer = recovery_enhancer_mixer_default<TSeq>`

- MixerFun< TSeq > **death_reduction_mixer** = death_reduction_mixer_default<TSeq>
- std::vector< epiworld_double > **array_double_tmp**
- std::vector< [Virus](#)< TSeq > * > **array_virus_tmp**
- virtual [Model](#)< TSeq > * **clone_ptr** ()

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

- **Model** ()
- **Model** (const [Model](#)< TSeq > &m)
- **Model** ([Model](#)< TSeq > &m)
- **Model** ([Model](#)< TSeq > &&m)
- [Model](#)< TSeq > & **operator=** (const [Model](#)< TSeq > &m)
- virtual ~**Model** ()

16.21.1 Detailed Description

```
template<typename TSeq>
class epiworld::Model< TSeq >
```

Core class of epiworld.

The model class provides the wrapper that puts together [Agent](#), [Virus](#), and [Tools](#).

Template Parameters

<i>TSeq</i>	Type of sequence. In principle, users can build models in which virus and human sequence is represented as numeric vectors (if needed.)
-------------	---

16.21.2 Member Function Documentation

16.21.2.1 add_globalevent()

```
template<typename TSeq >
void Model< TSeq >::add_globalevent (
    std::function< void(Model< TSeq > *)> fun,
    std::string name = "A global action",
    int date = -99 ) [inline]
```

Set a global action.

Parameters

<i>fun</i>	A function to be called on the prescribed date
<i>name</i>	Name of the action.
<i>date</i>	Integer indicating when the function is called (see details)

When date is less than zero, then the function is called at the end of every day. Otherwise, the function will be called only at the end of the indicated date.

16.21.2.2 clone_ptr()

```
template<typename TSeq >
Model< TSeq > * Model< TSeq >::clone_ptr [inline], [protected], [virtual]
```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

<i>copy</i>	
-------------	--

Reimplemented in [ModelSIRMixing< TSeq >](#), [ModelSIRLogit< TSeq >](#), [ModelSIRDCONN< TSeq >](#), [ModelSIRCONN< TSeq >](#), [ModelSEIRMixingQuarantine< TSeq >](#), [ModelSEIRMixing< TSeq >](#), [ModelSEIRDCONN< TSeq >](#), [ModelSEIRCONN< TSeq >](#), [ModelMeaslesMixing< TSeq >](#), [epiworld::epimodels::ModelMeaslesMixing< TSeq >](#), [epiworld::epimodels::ModelSEIRMixingQuara](#), [epiworld::epimodels::ModelMeaslesSchool< TSeq >](#), [epiworld::epimodels::ModelSIRMixing< TSeq >](#), [epiworld::epimodels::ModelSIRLogit< TSeq >](#), [epiworld::epimodels::ModelSEIRDCONN< TSeq >](#), [epiworld::epimodels::ModelSIRDCONN< TSeq >](#), [epiworld::epimodels::ModelSEIRCONN< TSeq >](#), and [epiworld::epimodels::ModelSIRCONN< TSeq >](#).

16.21.2.3 draw()

```
template<typename TSeq >
void Model< TSeq >::draw (
    DiagramType diagram_type = DiagramType::Mermaid,
    const std::string & fn_output = "",
    bool self = false ) [inline]
```

Draws a mermaid diagram of the model.

Parameters

<i>model</i>	The model to draw.
<i>fn_output</i>	The name of the file to write the diagram. If empty, the diagram will be printed to the standard output.
<i>self</i>	Whether to allow self-transitions.

16.21.2.4 events_add()

```
template<typename TSeq >
void Model< TSeq >::events_add (
    Agent< TSeq > * agent_,
    VirusPtr< TSeq > virus_,
    ToolPtr< TSeq > tool_,
    Entity< TSeq > * entity_,
    epiworld_fast_int new_state_,
    epiworld_fast_int queue_,
    EventFun< TSeq > call_,
    int idx_agent_,
    int idx_object_ ) [inline], [protected]
```

Construct a new [Event](#) object.

Parameters

<i>agent_</i>	Agent over which the action will be called
<i>virus_</i>	Virus pointer included in the action
<i>tool_</i>	Tool pointer included in the action
<i>entity_</i>	Entity pointer included in the action
<i>new_↔ state_</i>	New state of the agent
<i>call_</i>	Function the action will call
<i>queue_</i>	Change in the queue
<i>idx_↔ agent_</i>	Location of agent in object.
<i>idx_↔ object_</i>	Location of object in agent.

16.21.2.5 events_run()

```
template<typename TSeq >
void Model< TSeq >::events_run [inline]
```

Executes the stored action.

Parameters

<i>model↔ _</i>	Model over which it will be executed.
---------------------	---

16.21.2.6 load_agents_entities_ties()

```
template<typename TSeq >
void Model< TSeq >::load_agents_entities_ties (
    std::string fn,
    int skip ) [inline]
```

Associate agents-entities from a file.

The structure of the file should be two columns separated by space. The first column indexing between 0 and nagents-1, and the second column between 0 and nentities - 1.

Parameters

<i>fn</i>	Path to the file.
<i>skip</i>	How many rows to skip.

16.21.2.7 reset()

```
template<typename TSeq >
void Model< TSeq >::reset [inline], [virtual]
```

Reset the model.

Resetting the model will:

- clear the database
- restore the population (if `set_backup()` was called before)
- re-distribute tools
- re-distribute viruses
- set the date to 0

Reimplemented in [ModelSURV< TSeq >](#), [ModelSIRMixing< TSeq >](#), [ModelSIRLogit< TSeq >](#), [ModelSIRDCONN< TSeq >](#), [ModelSIRCONN< TSeq >](#), [ModelSEIRMixingQuarantine< TSeq >](#), [ModelSEIRMixing< TSeq >](#), [ModelSEIRDCONN< TSeq >](#), [ModelSEIRCONN< TSeq >](#), [ModelMeaslesMixing< TSeq >](#), [epiworld::epimodels::ModelMeaslesMixing< TSeq >](#), [epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >](#), [epiworld::epimodels::ModelMeaslesSchool< TSeq >](#), [epiworld::epimodels::ModelSIRMixing< TSeq >](#), [epiworld::epimodels::ModelSEIRMixing< TSeq >](#), [epiworld::epimodels::ModelSIRLogit< TSeq >](#), [epiworld::epimodels::ModelSIRDCONN< TSeq >](#), [epiworld::epimodels::ModelSIRCONN< TSeq >](#), [epiworld::epimodels::ModelSEIRDCONN< TSeq >](#), [epiworld::epimodels::ModelSEIRCONN< TSeq >](#), and [epiworld::epimodels::ModelSURV< TSeq >](#).

16.21.2.8 run_multiple()

```
template<typename TSeq >
void Model< TSeq >::run_multiple (
    epiworld_fast_uint ndays,
    epiworld_fast_uint nexperiments,
    int seed_ = -1,
    std::function< void(size_t, Model< TSeq > *)> fun = make_save_run<TSeq>(),
    bool reset = true,
    bool verbose = true,
    int nthreads = 1 ) [inline]
```

Parameters

<i>ndays</i>	Multiple runs of the simulation
--------------	---------------------------------

16.21.2.9 set_agents_data()

```
template<typename TSeq >
void Model< TSeq >::set_agents_data (
    double * data_,
    size_t ncols_ ) [inline]
```


Set the agents data object.

The data should be an array with the data stored in a column major order, i.e., by column.

Parameters

<i>data</i> ↔ _	Pointer to the first element of an array of size <code>size() * ncols_</code> .
<i>ncols</i> ↔ _	Number of features included in the data.

16.21.2.10 set_name()

```
template<typename TSeq >
void Model< TSeq >::set_name (
    std::string name ) [inline]
```

Set the name object.

Parameters

<i>name</i>	
-------------	--

16.21.2.11 write_data()

```
template<typename TSeq >
void Model< TSeq >::write_data (
    std::string fn_virus_info,
    std::string fn_virus_hist,
    std::string fn_tool_info,
    std::string fn_tool_hist,
    std::string fn_total_hist,
    std::string fn_transmission,
    std::string fn_transition,
    std::string fn_reproductive_number,
    std::string fn_generation_time ) const [inline]
```

Wrapper of `DataBase::write_data`

Parameters

<i>fn_virus_info</i>	Filename. Information about the virus.
<i>fn_virus_hist</i>	Filename. History of the virus.
<i>fn_tool_info</i>	Filename. Information about the tool.
<i>fn_tool_hist</i>	Filename. History of the tool.
<i>fn_total_hist</i>	Filename. Aggregated history (state)
<i>fn_transmission</i>	Filename. Transmission history.
<i>fn_transition</i>	Filename. Markov transition history.
<i>fn_reproductive_number</i>	Filename. Case by case reproductive number
<i>fn_generation_time</i>	

16.21.3 Member Data Documentation

16.21.3.1 initial_states_fun

```
template<typename TSeq >
std::function<void (Model<TSeq>*)> epiworld::Model< TSeq >::initial_states_fun [protected]
```

Initial value:

```
= [] (Model<TSeq> * )
    -> void {}
```

Function to distribute states. Goes along with the function

16.21.3.2 rbinomd

```
template<typename TSeq >
std::binomial_distribution epiworld::Model< TSeq >::rbinomd [protected]
```

Initial value:

```
=
    std::binomial_distribution<>()
```

16.21.3.3 rexp

```
template<typename TSeq >
std::exponential_distribution epiworld::Model< TSeq >::rexp [protected]
```

Initial value:

```
=
    std::exponential_distribution<>()
```

16.21.3.4 rgammad

```
template<typename TSeq >
std::gamma_distribution epiworld::Model< TSeq >::rgammad [protected]
```

Initial value:

```
=
    std::gamma_distribution<>()
```

16.21.3.5 rgeomd

```
template<typename TSeq >
std::geometric_distribution epiworld::Model< TSeq >::rgeomd [protected]
```

Initial value:

```
=
    std::geometric_distribution<>()
```

16.21.3.6 rlognormald

```
template<typename TSeq >
std::lognormal_distribution epiworld::Model< TSeq >::rlognormald [protected]
```

Initial value:

```
=
    std::lognormal_distribution<>()
```

16.21.3.7 rnbinomd

```
template<typename TSeq >
std::negative_binomial_distribution epiworld::Model< TSeq >::rnbinomd [protected]
```

Initial value:

```
=
    std::negative_binomial_distribution<>()
```

16.21.3.8 rnormd

```
template<typename TSeq >
std::normal_distribution epiworld::Model< TSeq >::rnormd [protected]
```

Initial value:

```
=
    std::normal_distribution<>(0.0)
```

16.21.3.9 rpoissd

```
template<typename TSeq >
std::poisson_distribution epiworld::Model< TSeq >::rpoissd [protected]
```

Initial value:

```
=
    std::poisson_distribution<>()
```

16.21.3.10 runifd

```
template<typename TSeq >
std::uniform_real_distribution<epiworld::Model< TSeq >::runifd [protected]
```

Initial value:

```
=
    std::uniform_real_distribution<> (0.0, 1.0)
```

16.21.3.11 time_elapsed

```
template<typename TSeq >
std::chrono::duration<epiworld_double, std::micro> epiworld::Model< TSeq >::time_elapsed [protected]
```

Initial value:

```
=
    std::chrono::duration<epiworld_double, std::micro>::zero()
```

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

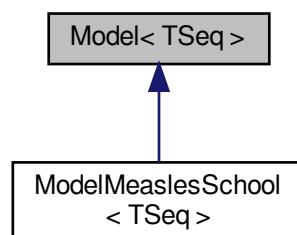
- epiworld.hpp

16.22 Model< TSeq > Class Template Reference

Core class of epiworld.

```
#include <model-bones.hpp>
```

Inheritance diagram for Model< TSeq >:



Collaboration diagram for Model< TSeq >:



Public Member Functions

- [DataBase< TSeq >](#) & **get_db** ()
- const [DataBase< TSeq >](#) & **get_db** () const
- epiworld_double & **operator()** (std::string pname)
- size_t **size** () const
- void [load_agents_entities_ties](#) (std::string fn, int skip)
 - Associate agents-entities from a file.*
- void [load_agents_entities_ties](#) (const std::vector< int > &agents_ids, const std::vector< int > &entities_ids)
 - Associate agents-entities from data.*
- void **load_agents_entities_ties** (const int *agents_id, const int *entities_id, size_t n)
- size_t [get_n_viruses](#) () const
 - Number of viruses in the model.*
- size_t [get_n_tools](#) () const
 - Number of tools in the model.*
- epiworld_fast_uint **get_ndays** () const
- epiworld_fast_uint **get_n_replicates** () const
- size_t **get_n_entities** () const
- void **set_ndays** (epiworld_fast_uint ndays)
- bool **get_verbose** () const
- [Model< TSeq >](#) & **verbose_off** ()
- [Model< TSeq >](#) & **verbose_on** ()
- int [today](#) () const
 - The current time of the model.*
- void [write_data](#) (std::string fn_virus_info, std::string fn_virus_hist, std::string fn_tool_info, std::string fn_tool_hist, std::string fn_total_hist, std::string fn_transmission, std::string fn_transition, std::string fn_reproductive_number, std::string fn_generation_time) const
 - Wrapper of DataBase::write_data*
- std::map< std::string, epiworld_double > & **params** ()
- virtual void [reset](#) ()
 - Reset the model.*
- const [Model< TSeq >](#) & **print** (bool lite=false) const
- void **get_elapsed** (std::string unit="auto", epiworld_double *last_elapsed=nullptr, epiworld_double *total_elapsed=nullptr, std::string *unit_abbr=nullptr, bool print=true) const
- void [add_globevent](#) (std::function< void([Model< TSeq >](#) *)> fun, std::string name="A global action", int date=-99)

Set a global action.

- void **add_globlevent** (GlobalEvent< TSeq > action)
- GlobalEvent< TSeq > & **get_globlevent** (std::string name)

Retrieve a global action by name.

- GlobalEvent< TSeq > & **get_globlevent** (size_t i)

Retrieve a global action by index.

- void **rm_globlevent** (std::string name)

Remove a global action by name.

- void **rm_globlevent** (size_t i)

Remove a global action by index.

- void **run_globlevents** ()
- void **clear_state_set** ()
- const std::vector< VirusPtr< TSeq > > & **get_viruses** () const
- const std::vector< ToolPtr< TSeq > > & **get_tools** () const
- Virus< TSeq > & **get_virus** (size_t id)
- Tool< TSeq > & **get_tool** (size_t id)
- void **set_agents_data** (double *data_, size_t ncols_)

Set the agents data object.

- double * **get_agents_data** ()
- size_t **get_agents_data_ncols** () const
- void **set_name** (std::string name)

Set the name object.

- std::string **get_name** () const
- bool **operator==** (const Model< TSeq > &other) const
- bool **operator!=** (const Model< TSeq > &other) const
- void **events_run** ()

Executes the stored action.

- void **draw** (DiagramType diagram_type=DiagramType::Mermaid, const std::string &fn_output="", bool self=false)

Draws a mermaid diagram of the model.

Set the backup object

backup can be used to restore the entire object after a run. This can be useful if the user wishes to have individuals start with the same network from the beginning.

- void **set_backup** ()

Random number generation

Parameters

eng	Random number generator
s	Seed

- void **set_rand_engine** (std::shared_ptr< std::mt19937 > &eng)
- std::shared_ptr< std::mt19937 > & **get_rand_endgine** ()
- void **seed** (size_t s)
- void **set_rand_norm** (epiworld_double mean, epiworld_double sd)
- void **set_rand_unif** (epiworld_double a, epiworld_double b)
- void **set_rand_exp** (epiworld_double lambda)
- void **set_rand_gamma** (epiworld_double alpha, epiworld_double beta)
- void **set_rand_lognormal** (epiworld_double mean, epiworld_double shape)
- void **set_rand_binom** (int n, epiworld_double p)

- void **set_rand_nbinom** (int n, epiworld_double p)
- void **set_rand_geom** (epiworld_double p)
- void **set_rand_pois** (epiworld_double lambda)
- epiworld_double **runif** ()
- epiworld_double **runif** (epiworld_double a, epiworld_double b)
- epiworld_double **rnorm** ()
- epiworld_double **rnorm** (epiworld_double mean, epiworld_double sd)
- epiworld_double **rgamma** ()
- epiworld_double **rgamma** (epiworld_double alpha, epiworld_double beta)
- epiworld_double **rexp** ()
- epiworld_double **rexp** (epiworld_double lambda)
- epiworld_double **rlognormal** ()
- epiworld_double **rlognormal** (epiworld_double mean, epiworld_double shape)
- int **rbinom** ()
- int **rbinom** (int n, epiworld_double p)
- int **rnbinom** ()
- int **rnbinom** (int n, epiworld_double p)
- int **rgeom** ()
- int **rgeom** (epiworld_double p)
- int **rpoiss** ()
- int **rpoiss** (epiworld_double lambda)

Add Virus/Tool to the model

This is done before the model has been initialized.

Parameters

v	<i>Virus to be added</i>
t	<i>Tool to be added</i>
preval	<i>Initial prevalence (initial state.) It can be specified as a proportion (between zero and one,) or an integer indicating number of individuals.</i>

- void **add_virus** ([Virus](#)< TSeq > &v)
- void **add_tool** ([Tool](#)< TSeq > &t)
- void **add_entity** ([Entity](#)< TSeq > e)
- void **rm_virus** (size_t virus_pos)
- void **rm_tool** (size_t tool_pos)
- void **rm_entity** (size_t entity_id)

Accessing population of the model

Parameters

fn	<i>std::string Filename of the edgelist file.</i>
skip	<i>int Number of lines to skip in fn.</i>
directed	<i>bool Whether the graph is directed or not.</i>
size	<i>Size of the network.</i>
al	<i>AdjList to read into the model.</i>

- void **agents_from_adjlist** (std::string fn, int size, int skip=0, bool directed=false)
- void **agents_from_edgelist** (const std::vector< int > &source, const std::vector< int > &target, int size, bool directed)
- void **agents_from_adjlist** ([AdjList](#) al)
- bool **is_directed** () const
- std::vector< [Agent](#)< TSeq > > & **get_agents** ()
Returns a reference to the vector of agents.

- [Agent](#)< TSeq > & **get_agent** (size_t i)
- std::vector< epiworld_fast_uint > [get_agents_states](#) () const
Returns a vector with the states of the agents.
- std::vector< [Viruses_const](#)< TSeq > > [get_agents_viruses](#) () const
Returns a const vector with the viruses of the agents.
- std::vector< [Viruses](#)< TSeq > > [get_agents_viruses](#) ()
Returns a vector with the viruses of the agents.
- std::vector< [Entity](#)< TSeq > > & **get_entities** ()
- [Entity](#)< TSeq > & **get_entity** (size_t entity_id, int *entity_pos=nullptr)
- [Model](#)< TSeq > & **agents_smallworld** (epiworld_fast_uint n=1000, epiworld_fast_uint k=5, bool d=false, epiworld_double p=.01)
- void **agents_empty_graph** (epiworld_fast_uint n=1000)

Functions to run the model

Parameters

seed	Seed to be used for Pseudo-RNG.
ndays	Number of days (steps) of the simulation.
fun	In the case of <i>run_multiple</i> , a function that is called after each experiment.

- void **update_state** ()
- void **mutate_virus** ()
- void **next** ()
- virtual [Model](#)< TSeq > & **run** (epiworld_fast_uint ndays, int seed=-1)
Runs the simulation (after initialization)
- void **run_multiple** (epiworld_fast_uint ndays, epiworld_fast_uint nexperiments, int seed_=-1, std::function< void(size_t, [Model](#)< TSeq > *)> fun=make_save_run< TSeq >(), bool **reset**=true, bool verbose=true, int nthreads=1)

Rewire the network preserving the degree sequence.

This implementation assumes an undirected network, thus if $\{(i,j), (k,l)\} \rightarrow \{(i,l), (k,j)\}$, the reciprocal is also true, i.e., $\{(j,i), (l,k)\} \rightarrow \{(j,k), (l,i)\}$.

Parameters

proportion	Proportion of ties to be rewired.
------------	-----------------------------------

Returns

A rewired version of the network.

- void **set_rewire_fun** (std::function< void(std::vector< [Agent](#)< TSeq > > *, [Model](#)< TSeq > *, epiworld_double)> fun)
- void **set_rewire_prop** (epiworld_double prop)
- epiworld_double **get_rewire_prop** () const
- void **rewire** ()

Export the network data in edgelist form

Parameters

fn	std::string. File name.
source	Integer vector
target	Integer vector

When passing the source and target, the function will write the edgelist on those.

- void **write_edgelist** (std::string fn) const
- void **write_edgelist** (std::vector< int > &source, std::vector< int > &target) const

Manage state (states) in the model

The functions `get_state` return the current values for the states included in the model.

Parameters

lab	<code>std::string</code> Name of the state.
-----	---

Returns

`add_state*` returns nothing.

`get_state_*` returns a vector of pairs with the states and their labels.

- void **add_state** (std::string lab, UpdateFun< TSeq > fun=nullptr)
- const std::vector< std::string > & **get_states** () const
- size_t **get_n_states** () const
- const std::vector< UpdateFun< TSeq > > & **get_state_fun** () const
- void **print_state_codes** () const

Initial states

These functions are called before the simulation starts.

Parameters

proportions↔	Vector of proportions for each state.
—	
queue_	Vector of queue for each state.

- virtual [Model](#)< TSeq > & **initial_states** (std::vector< double >, std::vector< int >)

Setting and accessing parameters from the model

[Tools](#) can incorporate parameters included in the model. Internally, parameters in the tool are stored as pointers to an `std::map<>` of parameters in the model. Using the `epiworld_fast_uint` method directly fetches the parameters in the order these were added to the tool. Accessing parameters via the `std::string` method involves searching the parameter directly in the `std::map<>` member of the model (so it is not recommended.)

The `par()` function members are aliases for `get_param()`.

In the case of the function `read_params`, users can pass a file listing parameters to be included in the model.

Each line in the file should have the following structure:

```
[name of parameter 1]: [value in double]
[name of parameter 2]: [value in double]
...
```

The only condition for parameter names is that these do not include a colon.

Parameters

initial_val	
pname	Name of the parameter to add or to fetch
fn	Path to the file containing parameters

Returns

The current value of the parameter in the model.

- `epiworld_double` **add_param** (`epiworld_double` initial_val, `std::string` pname, `bool` overwrite=false)
- `Model`< `TSeq` > & **read_params** (`std::string` fn, `bool` overwrite=false)
- `epiworld_double` **get_param** (`epiworld_fast_uint` k)
- `epiworld_double` **get_param** (`std::string` pname)
- `void` **set_param** (`std::string` pname, `epiworld_double` val)
- `epiworld_double` **par** (`std::string` pname) `const`

Set the user data object

Parameters

names	<i>string vector with the names of the variables.</i>
-------	---

- `void` **set_user_data** (`std::vector`< `std::string` > names)
- `[@`
- `void` **add_user_data** (`epiworld_fast_uint` j, `epiworld_double` x)
- `void` **add_user_data** (`std::vector`< `epiworld_double` > x)
- `UserData`< `TSeq` > & **get_user_data** ()

Queuing system

When queueing is on, the model will keep track of which agents are either in risk of exposure or exposed. This then is used at each step to act only on the aforementioned agents.

- `void` **queueing_on** ()
- *Activates the queueing system (default.)*
- `Model`< `TSeq` > & **queueing_off** ()
- *Deactivates the queueing system.*
- `bool` **is_queueing_on** () `const`
- *Query if the queueing system is on.*
- `Queue`< `TSeq` > & **get_queue** ()
- *Retrieve the `Queue` object.*

Get the susceptibility reduction object

Parameters

v	
---	--

Returns

epiworld_double

- `void` **set_susceptibility_reduction_mixer** (`MixerFun`< `TSeq` > fun)
- `void` **set_transmission_reduction_mixer** (`MixerFun`< `TSeq` > fun)
- `void` **set_recovery_enhancer_mixer** (`MixerFun`< `TSeq` > fun)
- `void` **set_death_reduction_mixer** (`MixerFun`< `TSeq` > fun)

Protected Member Functions

- `void` **dist_tools** ()
- `void` **dist_virus** ()

- void **dist_entities** ()
- void **chrono_start** ()
- void **chrono_end** ()
- void **events_add** (Agent< TSeq > *agent_, VirusPtr< TSeq > virus_, ToolPtr< TSeq > tool_, Entity< TSeq > *entity_, epiworld_fast_int new_state_, epiworld_fast_int queue_, EventFun< TSeq > call_, int idx_↵ agent_, int idx_object_)

Construct a new *Event* object.

Protected Attributes

- std::string **name** = ""
Name of the model.
- DataBase< TSeq > **db** = DataBase<TSeq>(*this)
- std::vector< Agent< TSeq > > **population** = {}
- bool **using_backup** = true
- std::vector< Agent< TSeq > > **population_backup** = {}
- bool **directed** = false
- std::vector< VirusPtr< TSeq > > **viruses** = {}
- std::vector< ToolPtr< TSeq > > **tools** = {}
- std::vector< Entity< TSeq > > **entities** = {}
- std::vector< Entity< TSeq > > **entities_backup** = {}
- std::shared_ptr< std::mt19937 > **engine** = std::make_shared< std::mt19937 >()
- std::uniform_real_distribution **runifd**
- std::normal_distribution **rnormd**
- std::gamma_distribution **rgammad**
- std::lognormal_distribution **rlognormald**
- std::exponential_distribution **rexp**
- std::binomial_distribution **rbinomd**
- std::negative_binomial_distribution **rnbinomd**
- std::geometric_distribution **rgeomd**
- std::poisson_distribution **rpoissd**
- std::function< void(std::vector< Agent< TSeq > > *, Model< TSeq > *, epiworld_double)> **rewire_fun**
- epiworld_double **rewire_prop** = 0.0
- std::map< std::string, epiworld_double > **parameters**
- epiworld_fast_uint **ndays** = 0
- **Progress pb**
- std::vector< UpdateFun< TSeq > > **state_fun** = {}
Functions to update states.
- std::vector< std::string > **states_labels** = {}
Labels of the states.
- std::function< void(Model< TSeq > *)> **initial_states_fun**
- epiworld_fast_uint **nstates** = 0u
- bool **verbose** = true
- int **current_date** = 0
- std::chrono::time_point< std::chrono::steady_clock > **time_start**
- std::chrono::time_point< std::chrono::steady_clock > **time_end**
- std::chrono::duration< epiworld_double, std::micro > **time_elapsed**
- epiworld_fast_uint **n_replicates** = 0u
- std::vector< GlobalEvent< TSeq > > **globalevents**
- **Queue< TSeq > queue**
- bool **use_queuing** = true
- std::vector< Event< TSeq > > **events** = {}
Variables used to keep track of the events to be made regarding viruses.

- `epiworld_fast_uint nactions = 0u`

Auxiliary variables for `AgentsSample<TSeq>` iterators

These variables+objects are used by the `AgentsSample<TSeq>` class for building efficient iterators over agents. The idea is to reduce the memory allocation, so only during the first call of `AgentsSample<TSeq>::AgentsSample(Model<TSeq>)` these vectors are allocated.

- `std::vector< Agent< TSeq > * > sampled_population`
- `size_t sampled_population_n = 0u`
- `std::vector< size_t > population_left`
- `size_t population_left_n = 0u`

Agents features

Optionally, a model can include an external data source pointing to agents information. The data can then be access through the `Agent::operator()` method.

- `double * agents_data = nullptr`
- `size_t agents_data_ncols = 0u`

Friends

- `class Agent< TSeq >`
- `class AgentsSample< TSeq >`
- `class DataBase< TSeq >`
- `class Queue< TSeq >`

Tool Mixers

These functions combine the effects tools have to deliver a single effect. For example, wearing a mask, been vaccinated, and the immune system combine together to jointly reduce the susceptibility for a given virus.

- `MixerFun< TSeq > susceptibility_reduction_mixer = susceptibility_reduction_mixer_default<TSeq>`
- `MixerFun< TSeq > transmission_reduction_mixer = transmission_reduction_mixer_default<TSeq>`
- `MixerFun< TSeq > recovery_enhancer_mixer = recovery_enhancer_mixer_default<TSeq>`
- `MixerFun< TSeq > death_reduction_mixer = death_reduction_mixer_default<TSeq>`
- `std::vector< epiworld_double > array_double_tmp`
- `std::vector< Virus< TSeq > * > array_virus_tmp`
- `virtual Model< TSeq > * clone_ptr ()`

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

- `Model ()`
- `Model (const Model< TSeq > &m)`
- `Model (Model< TSeq > &m)`
- `Model (Model< TSeq > &&m)`
- `Model< TSeq > & operator= (const Model< TSeq > &m)`
- `virtual ~Model ()`

16.22.1 Detailed Description

```
template<typename TSeq>
class Model< TSeq >
```

Core class of epiworld.

The model class provides the wrapper that puts together `Agent`, `Virus`, and `Tools`.

Template Parameters

<i>TSeq</i>	Type of sequence. In principle, users can build models in which virus and human sequence is represented as numeric vectors (if needed.)
-------------	---

16.22.2 Member Function Documentation

16.22.2.1 add_globalevent()

```
template<typename TSeq >
void Model< TSeq >::add_globalevent (
    std::function< void(Model< TSeq > *)> fun,
    std::string name = "A global action",
    int date = -99 ) [inline]
```

Set a global action.

Parameters

<i>fun</i>	A function to be called on the prescribed date
<i>name</i>	Name of the action.
<i>date</i>	Integer indicating when the function is called (see details)

When date is less than zero, then the function is called at the end of every day. Otherwise, the function will be called only at the end of the indicated date.

16.22.2.2 clone_ptr()

```
template<typename TSeq >
Model< TSeq > * Model< TSeq >::clone_ptr [inline], [protected], [virtual]
```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

<i>copy</i>	
-------------	--

Reimplemented in [ModelMeaslesSchool< TSeq >](#).

16.22.2.3 draw()

```
template<typename TSeq >
void Model< TSeq >::draw (
```

```
DiagramType diagram_type = DiagramType::Mermaid,
const std::string & fn_output = "",
bool self = false ) [inline]
```

Draws a mermaid diagram of the model.

Parameters

<i>model</i>	The model to draw.
<i>fn_output</i>	The name of the file to write the diagram. If empty, the diagram will be printed to the standard output.
<i>self</i>	Whether to allow self-transitions.

16.22.2.4 events_add()

```
template<typename TSeq >
void Model< TSeq >::events_add (
    Agent< TSeq > * agent_,
    VirusPtr< TSeq > virus_,
    ToolPtr< TSeq > tool_,
    Entity< TSeq > * entity_,
    epiworld_fast_int new_state_,
    epiworld_fast_int queue_,
    EventFun< TSeq > call_,
    int idx_agent_,
    int idx_object_ ) [inline], [protected]
```

Construct a new [Event](#) object.

Parameters

<i>agent_</i>	Agent over which the action will be called
<i>virus_</i>	Virus pointer included in the action
<i>tool_</i>	Tool pointer included in the action
<i>entity_</i>	Entity pointer included in the action
<i>new_↔ state_</i>	New state of the agent
<i>call_</i>	Function the action will call
<i>queue_</i>	Change in the queue
<i>idx_↔ agent_</i>	Location of agent in object.
<i>idx_↔ object_</i>	Location of object in agent.

16.22.2.5 events_run()

```
template<typename TSeq >
void Model< TSeq >::events_run [inline]
```

Executes the stored action.

Parameters

<i>model</i> ↔	Model over which it will be executed.
—	

16.22.2.6 load_agents_entities_ties()

```
template<typename TSeq >
void Model< TSeq >::load_agents_entities_ties (
    std::string fn,
    int skip ) [inline]
```

Associate agents-entities from a file.

The structure of the file should be two columns separated by space. The first column indexing between 0 and nagents-1, and the second column between 0 and nentities - 1.

Parameters

<i>fn</i>	Path to the file.
<i>skip</i>	How many rows to skip.

16.22.2.7 reset()

```
template<typename TSeq >
void Model< TSeq >::reset [inline], [virtual]
```

Reset the model.

Resetting the model will:

- clear the database
- restore the population (if `set_backup()` was called before)
- re-distribute tools
- re-distribute viruses
- set the date to 0

Reimplemented in [ModelMeaslesSchool< TSeq >](#).

16.22.2.8 run_multiple()

```
template<typename TSeq >
void Model< TSeq >::run_multiple (
    epiworld_fast_uint ndays,
    epiworld_fast_uint nexperiments,
    int seed_ = -1,
    std::function< void(size_t, Model< TSeq > *)> fun = make_save_run<TSeq>(),
    bool reset = true,
    bool verbose = true,
    int nthreads = 1 ) [inline]
```

Parameters

<i>ndays</i>	Multiple runs of the simulation
--------------	---------------------------------

16.22.2.9 set_agents_data()

```
template<typename TSeq >
void Model< TSeq >::set_agents_data (
    double * data_,
    size_t ncols_ ) [inline]
```

Set the agents data object.

The data should be an array with the data stored in a column major order, i.e., by column.

Parameters

<i>data</i> ↔ —	Pointer to the first element of an array of size <code>size() * ncols_</code> .
<i>ncols</i> ↔ —	Number of features included in the data.

16.22.2.10 set_name()

```
template<typename TSeq >
void Model< TSeq >::set_name (
    std::string name ) [inline]
```

Set the name object.

Parameters

<i>name</i>	
-------------	--

16.22.2.11 write_data()

```
template<typename TSeq >
void Model< TSeq >::write_data (
    std::string fn_virus_info,
    std::string fn_virus_hist,
    std::string fn_tool_info,
    std::string fn_tool_hist,
    std::string fn_total_hist,
    std::string fn_transmission,
    std::string fn_transition,
    std::string fn_reproductive_number,
    std::string fn_generation_time ) const [inline]
```

Wrapper of DataBase::write_data

Parameters

<i>fn_virus_info</i>	Filename. Information about the virus.
<i>fn_virus_hist</i>	Filename. History of the virus.
<i>fn_tool_info</i>	Filename. Information about the tool.
<i>fn_tool_hist</i>	Filename. History of the tool.
<i>fn_total_hist</i>	Filename. Aggregated history (state)
<i>fn_transmission</i>	Filename. Transmission history.
<i>fn_transition</i>	Filename. Markov transition history.
<i>fn_reproductive_number</i>	Filename. Case by case reproductive number

16.22.3 Member Data Documentation

16.22.3.1 initial_states_fun

```
template<typename TSeq >
std::function<void (Model<TSeq>*)> Model< TSeq >::initial_states_fun [protected]
```

Initial value:

```
= [] (Model<TSeq> * )
    -> void {}
```

Function to distribute states. Goes along with the function

16.22.3.2 rbinomd

```
template<typename TSeq >
std::binomial_distribution Model< TSeq >::rbinomd [protected]
```

Initial value:

```
=
    std::binomial_distribution<>()
```

16.22.3.3 rexp

```
template<typename TSeq >  
std::exponential_distribution Model< TSeq >::rexp [protected]
```

Initial value:

```
=  
    std::exponential_distribution<>()
```

16.22.3.4 rgammad

```
template<typename TSeq >  
std::gamma_distribution Model< TSeq >::rgammad [protected]
```

Initial value:

```
=  
    std::gamma_distribution<>()
```

16.22.3.5 rgeomd

```
template<typename TSeq >  
std::geometric_distribution Model< TSeq >::rgeomd [protected]
```

Initial value:

```
=  
    std::geometric_distribution<>()
```

16.22.3.6 rlognormald

```
template<typename TSeq >  
std::lognormal_distribution Model< TSeq >::rlognormald [protected]
```

Initial value:

```
=  
    std::lognormal_distribution<>()
```

16.22.3.7 rnbinomd

```
template<typename TSeq >  
std::negative_binomial_distribution Model< TSeq >::rnbinomd [protected]
```

Initial value:

```
=  
    std::negative_binomial_distribution<>()
```

16.22.3.8 rnormd

```
template<typename TSeq >
std::normal_distribution Model< TSeq >::rnormd [protected]
```

Initial value:

```
=
    std::normal_distribution<>(0.0)
```

16.22.3.9 rpoissd

```
template<typename TSeq >
std::poisson_distribution Model< TSeq >::rpoissd [protected]
```

Initial value:

```
=
    std::poisson_distribution<>()
```

16.22.3.10 runifd

```
template<typename TSeq >
std::uniform_real_distribution Model< TSeq >::runifd [protected]
```

Initial value:

```
=
    std::uniform_real_distribution<>(0.0, 1.0)
```

16.22.3.11 time_elapsed

```
template<typename TSeq >
std::chrono::duration<epiworld_double, std::micro> Model< TSeq >::time_elapsed [protected]
```

Initial value:

```
=
    std::chrono::duration<epiworld_double, std::micro>::zero()
```

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

- include/epiworld/agent-bones.hpp
- include/epiworld/model-bones.hpp
- include/epiworld/model-meat-print.hpp
- include/epiworld/model-meat.hpp

16.23 epiworld::ModelDiagram Class Reference

Public Member Functions

- void **draw_from_data** (DiagramType diagram_type, const std::vector< std::string > &states, const std::vector< epiworld_double > &tprob, const std::string &fn_output="", bool self=false)
- void **draw_from_file** (DiagramType diagram_type, const std::string &fn_transition, const std::string &fn_output="", bool self=false)
- void **draw_from_files** (DiagramType diagram_type, const std::vector< std::string > &fns_transition, const std::string &fn_output="", bool self=false)

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

- epiworld.hpp

16.24 ModelDiagram Class Reference

Public Member Functions

- void **draw_from_data** (DiagramType diagram_type, const std::vector< std::string > &states, const std::vector< epiworld_double > &tprob, const std::string &fn_output="", bool self=false)
- void **draw_from_file** (DiagramType diagram_type, const std::string &fn_transition, const std::string &fn_output="", bool self=false)
- void **draw_from_files** (DiagramType diagram_type, const std::vector< std::string > &fns_transition, const std::string &fn_output="", bool self=false)

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

- include/epiworld/modeldiagram-bones.hpp
- include/epiworld/modeldiagram-meat.hpp

16.25 epiworld::epimodels::ModelDiffNet< TSeq > Class Template Reference

Template for a [Network](#) Diffusion [Model](#).

```
#include <epiworld.hpp>
```

Inheritance diagram for epiworld::epimodels::ModelDiffNet< TSeq >:



Collaboration diagram for `epiworld::epimodels::ModelDiffNet< TSeq >`:



Public Member Functions

- **ModelDiffNet** ([ModelDiffNet](#)< TSeq > &model, const std::string &innovation_name, epiworld_double prevalence, epiworld_double prob_adopt, bool normalize_exposure=true, double *agents_data=nullptr, size_t data_ncols=0u, std::vector< size_t > data_cols={}, std::vector< double > params={})
- **ModelDiffNet** (const std::string &innovation_name, epiworld_double prevalence, epiworld_double prob_adopt, bool normalize_exposure=true, double *agents_data=nullptr, size_t data_ncols=0u, std::vector< size_t > data_cols={}, std::vector< double > params={})

Public Attributes

- bool **normalize_exposure** = true
- std::vector< size_t > **data_cols**
- std::vector< double > **params**

Static Public Attributes

- static const int **NONADOPTER** = 0
- static const int **ADOPTER** = 1

Additional Inherited Members

16.25.1 Detailed Description

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
class epiworld::epimodels::ModelDiffNet< TSeq >
```

Template for a [Network](#) Diffusion [Model](#).

Parameters

<i>model</i>	A Model<TSeq> object where to set up the SIR.
<i>vname</i>	std::string Name of the virus
<i>initial_prevalence</i>	epiworld_double Initial prevalence
<i>initial_efficacy</i>	epiworld_double Initial susceptibility_reduction of the immune system
<i>initial_recovery</i>	epiworld_double Initial recovery rate of the immune system

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

- epiworld.hpp

16.26 ModelDiffNet< TSeq > Class Template Reference

Template for a [Network](#) Diffusion [Model](#).

```
#include <diffnet.hpp>
```

Inheritance diagram for ModelDiffNet< TSeq >:



Collaboration diagram for ModelDiffNet< TSeq >:



Public Member Functions

- **ModelDiffNet** ([ModelDiffNet](#)< TSeq > &model, const std::string &innovation_name, epiworld_double prevalence, epiworld_double prob_adopt, bool normalize_exposure=true, double *agents_data=nullptr, size_t data_ncols=0u, std::vector< size_t > data_cols={}, std::vector< double > params={})
- **ModelDiffNet** (const std::string &innovation_name, epiworld_double prevalence, epiworld_double prob_adopt, bool normalize_exposure=true, double *agents_data=nullptr, size_t data_ncols=0u, std::vector< size_t > data_cols={}, std::vector< double > params={})

Public Attributes

- bool **normalize_exposure** = true
- std::vector< size_t > **data_cols**
- std::vector< double > **params**

Static Public Attributes

- static const int **NONADOPTER** = 0
- static const int **ADOPTER** = 1

Additional Inherited Members

16.26.1 Detailed Description

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
class ModelDiffNet< TSeq >
```

Template for a [Network](#) Diffusion [Model](#).

Parameters

<i>model</i>	A Model<TSeq> object where to set up the SIR.
<i>vname</i>	std::string Name of the virus
<i>initial_prevalence</i>	epiworld_double Initial prevalence
<i>initial_efficiency</i>	epiworld_double Initial susceptibility_reduction of the immune system
<i>initial_recovery</i>	epiworld_double Initial recovery rate of the immune system

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

- include/epiworld/models/diffnet.hpp

16.27 epiworld::epimodels::ModelMeaslesMixing< TSeq > Class Template Reference

Measles model with population mixing, quarantine, and contact tracing.

```
#include <epiworld.hpp>
```


Inheritance diagram for epiworld::epimodels::ModelMeaslesMixing< TSeq >:



Collaboration diagram for epiworld::epimodels::ModelMeaslesMixing< TSeq >:



Public Member Functions

- [ModelMeaslesMixing](#) ([ModelMeaslesMixing](#)< TSeq > &model, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double vax_↔ efficacy, epiworld_double vax_reduction_recovery_rate, epiworld_double incubation_period, epiworld_double prodromal_period, epiworld_double rash_period, std::vector< double > contact_matrix, epiworld_double hospitalization_rate, epiworld_double hospitalization_period, epiworld_double days_undetected, epiworld_↔ _fast_int quarantine_period, epiworld_double quarantine_willingness, epiworld_double isolation_willingness, epiworld_fast_int isolation_period, epiworld_double prop_vaccinated, epiworld_double contact_tracing_↔ success_rate=1.0, epiworld_fast_uint contact_tracing_days_prior=4u)
Constructs a [ModelMeaslesMixing](#) object.
- [ModelMeaslesMixing](#) (epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double vax_efficacy, epiworld_double vax_reduction_↔ recovery_rate, epiworld_double incubation_period, epiworld_double prodromal_period, epiworld_double rash_period, std::vector< double > contact_matrix, epiworld_double hospitalization_rate, epiworld_double hospitalization_period, epiworld_double days_undetected, epiworld_fast_int quarantine_period, epiworld_↔ _double quarantine_willingness, epiworld_double isolation_willingness, epiworld_fast_int isolation_period,

epiworld_double prop_vaccinated, epiworld_double contact_tracing_success_rate=1.0, epiworld_fast_uint contact_tracing_days_prior=4u)

Constructs a [ModelMeaslesMixing](#) object.

- [ModelMeaslesMixing](#)< TSeq > & [run](#) (epiworld_fast_uint ndays, int seed=-1)

Run the model simulation.

- void [reset](#) ()

Reset the model to initial state.

- [Model](#)< TSeq > * [clone_ptr](#) ()

Create a clone of this model.

- [ModelMeaslesMixing](#)< TSeq > & [initial_states](#) (std::vector< double > proportions_, std::vector< int > queue_={})

Set the initial states of the model.

- void [set_contact_matrix](#) (std::vector< double > cmat)

Set the contact matrix for population mixing.

- std::vector< double > [get_contact_matrix](#) () const

Get the current contact matrix.

- std::vector< size_t > [get_agent_quarantine_triggered](#) () const

Get the quarantine trigger status for all agents.

- std::vector< bool > [get_quarantine_willingness](#) () const

Get the quarantine willingness for all agents.

- std::vector< bool > [get_isolation_willingness](#) () const

Get the isolation willingness for all agents.

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **EXPOSED** = 1
- static const int **PRODROMAL** = 2
- static const int **RASH** = 3
- static const int **ISOLATED** = 4
- static const int **ISOLATED_RECOVERED** = 5
- static const int **DETECTED_HOSPITALIZED** = 6
- static const int **QUARANTINED_EXPOSED** = 7
- static const int **QUARANTINED_SUSCEPTIBLE** = 8
- static const int **QUARANTINED_PRODROMAL** = 9
- static const int **QUARANTINED_RECOVERED** = 10
- static const int **HOSPITALIZED** = 11
- static const int **RECOVERED** = 12
- static const size_t **QUARANTINE_PROCESS_INACTIVE** = 0u
- static const size_t **QUARANTINE_PROCESS_ACTIVE** = 1u
- static const size_t **QUARANTINE_PROCESS_DONE** = 2u

Additional Inherited Members

16.27.1 Detailed Description

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
class epiworld::epimodels::ModelMeaslesMixing< TSeq >
```

Measles model with population mixing, quarantine, and contact tracing.

This class implements a Measles epidemiological model based on the SEIR framework with additional features including:

- Population mixing based on contact matrices
- Measles-specific disease progression: Susceptible → Exposed → Prodromal → Rash
- Prodromal individuals are infectious (replace the "Infected" state in SEIR)
- Rash individuals are no longer infectious but can be detected for isolation
- Quarantine measures for exposed contacts during contact tracing
- Isolation policies for detected individuals during the rash state
- Contact tracing with configurable success rates
- Hospitalization of severe cases
- Individual willingness to comply with public health measures

The model supports 13 distinct states:

- Susceptible: Individuals who can become infected
- Exposed: Infected but not yet infectious (incubation period)
- Prodromal: Infectious individuals in the community (replaces "Infected" in SEIR)
- Rash: Non-infectious individuals with visible symptoms (detection occurs here)
- Isolated: Detected individuals in self-isolation
- Isolated Recovered: Recovered individuals still in isolation
- Detected Hospitalized: Hospitalized individuals who were contact-traced
- Quarantined Exposed: Exposed individuals in quarantine due to contact tracing
- Quarantined Susceptible: Susceptible individuals in quarantine due to contact tracing
- Quarantined Prodromal: Prodromal individuals in quarantine due to contact tracing
- Quarantined Recovered: Recovered individuals in quarantine due to contact tracing
- Hospitalized: Individuals requiring hospital care
- Recovered: Individuals who have recovered and gained immunity

Template Parameters

<code>TSeq</code>	Type for genetic sequences (default: <code>EPI_DEFAULT_TSEQ</code>)
-------------------	--

16.27.2 Constructor & Destructor Documentation

16.27.2.1 `ModelMeaslesMixing()` [1/2]

```
template<typename TSeq >
ModelMeaslesMixing< TSeq >::ModelMeaslesMixing (
```

```

ModelMeaslesMixing< TSeq > & model,
epiworld_fast_uint n,
epiworld_double prevalence,
epiworld_double contact_rate,
epiworld_double transmission_rate,
epiworld_double vax_efficacy,
epiworld_double vax_reduction_recovery_rate,
epiworld_double incubation_period,
epiworld_double prodromal_period,
epiworld_double rash_period,
std::vector< double > contact_matrix,
epiworld_double hospitalization_rate,
epiworld_double hospitalization_period,
epiworld_double days_undetected,
epiworld_fast_int quarantine_period,
epiworld_double quarantine_willingness,
epiworld_double isolation_willingness,
epiworld_fast_int isolation_period,
epiworld_double prop_vaccinated,
epiworld_double contact_tracing_success_rate = 1.0,
epiworld_fast_uint contact_tracing_days_prior = 4u ) [inline]

```

Constructs a [ModelMeaslesMixing](#) object.

Template for a Measles model with population mixing, quarantine, and contact tracing.

Parameters

<i>model</i>	A reference to an existing ModelMeaslesMixing object.
<i>n</i>	The number of entities in the model.
<i>prevalence</i>	The initial prevalence of the disease in the model.
<i>contact_rate</i>	The contact rate between entities in the model.
<i>transmission_rate</i>	The transmission rate of the disease in the model.
<i>vax_efficacy</i>	The efficacy of the vaccine.
<i>vax_reduction_recovery_rate</i>	The reduction in recovery rate due to the vaccine.
<i>incubation_period</i>	The incubation period of the disease in the model.
<i>prodromal_period</i>	The prodromal period of the disease in the model.
<i>rash_period</i>	The rash period of the disease in the model.
<i>contact_matrix</i>	The contact matrix between entities in the model. Specified in column-major order.
<i>hospitalization_rate</i>	The rate at which infected individuals are hospitalized.
<i>hospitalization_period</i>	The average duration of hospitalization in days.
<i>days_undetected</i>	The average number of days an infected individual remains undetected.
<i>quarantine_period</i>	The duration of quarantine in days for exposed contacts.
<i>quarantine_willingness</i>	The proportion of individuals willing to comply with quarantine measures.
<i>isolation_willingness</i>	The proportion of individuals willing to self-isolate when detected.
<i>isolation_period</i>	The duration of isolation in days for detected infected individuals.
<i>prop_vaccinated</i>	The proportion of vaccinated agents.
<i>contact_tracing_success_rate</i>	The probability of successfully identifying and tracing contacts (default: 1.0).
<i>contact_tracing_days_prior</i>	The number of days prior to detection for which contacts are traced (default: 4).
<i>model</i>	A ModelMeaslesMixing<TSeq> object where to set up the model.
<i>n</i>	Number of agents in the population
<i>prevalence</i>	Initial prevalence (proportion of infected individuals)

Parameters

<i>contact_rate</i>	Average number of contacts (interactions) per step
<i>transmission_rate</i>	Probability of transmission per contact
<i>vax_efficacy</i>	The efficacy of the vaccine
<i>vax_reduction_recovery_rate</i>	The reduction in recovery rate due to the vaccine
<i>incubation_period</i>	Average incubation period in days
<i>prodromal_period</i>	Average prodromal period in days
<i>rash_period</i>	Average rash period in days
<i>contact_matrix</i>	Contact matrix specifying mixing patterns between population groups
<i>hospitalization_rate</i>	Rate at which infected individuals are hospitalized
<i>hospitalization_period</i>	Average duration of hospitalization in days
<i>days_undetected</i>	Average number of days an infected individual remains undetected
<i>quarantine_period</i>	Duration of quarantine in days for exposed contacts
<i>quarantine_willingness</i>	Proportion of individuals willing to comply with quarantine
<i>isolation_willingness</i>	Proportion of individuals willing to self-isolate when detected
<i>isolation_period</i>	Duration of isolation in days for detected infected individuals
<i>prop_vaccinated</i>	Proportion of vaccinated agents
<i>contact_tracing_success_rate</i>	Probability of successfully identifying contacts during tracing
<i>contact_tracing_days_prior</i>	Number of days prior to detection for contact tracing

16.27.2.2 `ModelMeaslesMixing()` [2/2]

```
template<typename TSeq >
ModelMeaslesMixing< TSeq >::ModelMeaslesMixing (
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double vax_efficacy,
    epiworld_double vax_reduction_recovery_rate,
    epiworld_double incubation_period,
    epiworld_double prodromal_period,
    epiworld_double rash_period,
    std::vector< double > contact_matrix,
    epiworld_double hospitalization_rate,
    epiworld_double hospitalization_period,
    epiworld_double days_undetected,
    epiworld_fast_int quarantine_period,
    epiworld_double quarantine_willingness,
    epiworld_double isolation_willingness,
    epiworld_fast_int isolation_period,
    epiworld_double prop_vaccinated,
    epiworld_double contact_tracing_success_rate = 1.0,
    epiworld_fast_uint contact_tracing_days_prior = 4u ) [inline]
```

Constructs a `ModelMeaslesMixing` object.

Parameters

<i>n</i>	The number of entities in the model.
<i>prevalence</i>	The initial prevalence of the disease in the model.
<i>contact_rate</i>	The contact rate between entities in the model.
<i>transmission_rate</i>	The transmission rate of the disease in the model.
<i>vax_efficacy</i>	The efficacy of the vaccine.
<i>vax_reduction_recovery_rate</i>	The reduction in recovery rate due to the vaccine.
<i>incubation_period</i>	The incubation period of the disease in the model.
<i>prodromal_period</i>	The prodromal period of the disease in the model.
<i>rash_period</i>	The rash period of the disease in the model.
<i>contact_matrix</i>	The contact matrix between entities in the model.
<i>hospitalization_rate</i>	The rate at which infected individuals are hospitalized.
<i>hospitalization_period</i>	The average duration of hospitalization in days.
<i>days_undetected</i>	The average number of days an infected individual remains undetected.
<i>quarantine_period</i>	The duration of quarantine in days for exposed contacts.
<i>quarantine_willingness</i>	The proportion of individuals willing to comply with quarantine measures.
<i>isolation_willingness</i>	The proportion of individuals willing to self-isolate when detected.
<i>isolation_period</i>	The duration of isolation in days for detected infected individuals.
<i>prop_vaccinated</i>	The proportion of vaccinated agents.
<i>contact_tracing_success_rate</i>	The probability of successfully identifying and tracing contacts (default: 1.0).
<i>contact_tracing_days_prior</i>	The number of days prior to detection for which contacts are traced (default: 4).

16.27.3 Member Function Documentation

16.27.3.1 clone_ptr()

```
template<typename TSeq >
Model< TSeq > * ModelMeaslesMixing< TSeq >::clone_ptr [inline], [virtual]
```

Create a clone of this model.

Returns

Pointer to a new model instance with the same configuration

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.27.3.2 get_agent_quarantine_triggered()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
std::vector< size_t > epiworld::epimodels::ModelMeaslesMixing< TSeq >::get_agent_quarantine↔
_triggered ( ) const [inline]
```

Get the quarantine trigger status for all agents.

Returns

Vector indicating quarantine process status for each agent

16.27.3.3 get_contact_matrix()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
std::vector< double > epiworld::epimodels::ModelMeaslesMixing< TSeq >::get_contact_matrix ( )
const [inline]
```

Get the current contact matrix.

Returns

Vector representing the contact matrix

16.27.3.4 get_isolation_willingness()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
std::vector< bool > epiworld::epimodels::ModelMeaslesMixing< TSeq >::get_isolation_willingness
( ) const [inline]
```

Get the isolation willingness for all agents.

Returns

Vector of boolean values indicating each agent's willingness to self-isolate

16.27.3.5 get_quarantine_willingness()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
std::vector< bool > epiworld::epimodels::ModelMeaslesMixing< TSeq >::get_quarantine_willingness
( ) const [inline]
```

Get the quarantine willingness for all agents.

Returns

Vector of boolean values indicating each agent's willingness to quarantine

16.27.3.6 initial_states()

```
template<typename TSeq >
ModelMeaslesMixing< TSeq > & ModelMeaslesMixing< TSeq >::initial_states (
    std::vector< double > proportions_,
    std::vector< int > queue_ = {} ) [inline], [virtual]
```

Set the initial states of the model.

Parameters

<i>proportions</i> ↔ —	Double vector with two elements: <ul style="list-style-type: none"> • [0]: The proportion of initially infected individuals who start in the exposed state. • [1]: The proportion of initially non-infected individuals who have recovered (immune).
<i>queue_</i>	Optional vector for queuing specifications (default: empty).

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.27.3.7 run()

```
template<typename TSeq >
ModelMeaslesMixing< TSeq > & ModelMeaslesMixing< TSeq >::run (
    epiworld_fast_uint ndays,
    int seed = -1 ) [inline], [virtual]
```

Run the model simulation.

Parameters

<i>ndays</i>	Number of days to simulate
<i>seed</i>	Random seed for reproducibility (default: -1 for random seed)

Returns

Reference to this model instance

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.27.3.8 set_contact_matrix()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
void epiworld::epimodels::ModelMeaslesMixing< TSeq >::set_contact_matrix (
    std::vector< double > cmat ) [inline]
```

Set the contact matrix for population mixing.

Parameters

<i>cmat</i>	Contact matrix specifying interaction rates between groups
-------------	--

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

- epiworld.hpp

16.28 ModelMeaslesMixing< TSeq > Class Template Reference

Measles model with population mixing, quarantine, and contact tracing.

```
#include <measlesmixing.hpp>
```

Inheritance diagram for ModelMeaslesMixing< TSeq >:



Collaboration diagram for ModelMeaslesMixing< TSeq >:



Public Member Functions

- [ModelMeaslesMixing](#) ([ModelMeaslesMixing](#)< TSeq > &model, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double vax_↵ efficacy, epiworld_double vax_reduction_recovery_rate, epiworld_double incubation_period, epiworld_double prodromal_period, epiworld_double rash_period, std::vector< double > contact_matrix, epiworld_double hospitalization_rate, epiworld_double hospitalization_period, epiworld_double days_undetected, epiworld_↵ _fast_int quarantine_period, epiworld_double quarantine_willingness, epiworld_double isolation_willingness, epiworld_fast_int isolation_period, epiworld_double prop_vaccinated, epiworld_double contact_tracing_↵ success_rate=1.0, epiworld_fast_uint contact_tracing_days_prior=4u)

Constructs a [ModelMeaslesMixing](#) object.

- [ModelMeaslesMixing](#) (epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double vax_efficacy, epiworld_double vax_reduction_recovery_rate, epiworld_double incubation_period, epiworld_double prodromal_period, epiworld_double rash_period, std::vector< double > contact_matrix, epiworld_double hospitalization_rate, epiworld_double hospitalization_period, epiworld_double days_undetected, epiworld_fast_int quarantine_period, epiworld_double quarantine_willingness, epiworld_double isolation_willingness, epiworld_fast_int isolation_period, epiworld_double prop_vaccinated, epiworld_double contact_tracing_success_rate=1.0, epiworld_fast_int contact_tracing_days_prior=4u)
Constructs a [ModelMeaslesMixing](#) object.
- [ModelMeaslesMixing](#)< TSeq > & [run](#) (epiworld_fast_uint ndays, int seed=-1)
Run the model simulation.
- void [reset](#) ()
Reset the model to initial state.
- [Model](#)< TSeq > * [clone_ptr](#) ()
Create a clone of this model.
- [ModelMeaslesMixing](#)< TSeq > & [initial_states](#) (std::vector< double > proportions_, std::vector< int > queue_={})
Set the initial states of the model.
- void [set_contact_matrix](#) (std::vector< double > cmat)
Set the contact matrix for population mixing.
- std::vector< double > [get_contact_matrix](#) () const
Get the current contact matrix.
- std::vector< size_t > [get_agent_quarantine_triggered](#) () const
Get the quarantine trigger status for all agents.
- std::vector< bool > [get_quarantine_willingness](#) () const
Get the quarantine willingness for all agents.
- std::vector< bool > [get_isolation_willingness](#) () const
Get the isolation willingness for all agents.

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **EXPOSED** = 1
- static const int **PRODROMAL** = 2
- static const int **RASH** = 3
- static const int **ISOLATED** = 4
- static const int **ISOLATED_RECOVERED** = 5
- static const int **DETECTED_HOSPITALIZED** = 6
- static const int **QUARANTINED_EXPOSED** = 7
- static const int **QUARANTINED_SUSCEPTIBLE** = 8
- static const int **QUARANTINED_PRODROMAL** = 9
- static const int **QUARANTINED_RECOVERED** = 10
- static const int **HOSPITALIZED** = 11
- static const int **RECOVERED** = 12
- static const size_t **QUARANTINE_PROCESS_INACTIVE** = 0u
- static const size_t **QUARANTINE_PROCESS_ACTIVE** = 1u
- static const size_t **QUARANTINE_PROCESS_DONE** = 2u

Additional Inherited Members

16.28.1 Detailed Description

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
class ModelMeaslesMixing< TSeq >
```

Measles model with population mixing, quarantine, and contact tracing.

This class implements a Measles epidemiological model based on the SEIR framework with additional features including:

- Population mixing based on contact matrices
- Measles-specific disease progression: Susceptible → Exposed → Prodromal → Rash
- Prodromal individuals are infectious (replace the "Infected" state in SEIR)
- Rash individuals are no longer infectious but can be detected for isolation
- Quarantine measures for exposed contacts during contact tracing
- Isolation policies for detected individuals during the rash state
- Contact tracing with configurable success rates
- Hospitalization of severe cases
- Individual willingness to comply with public health measures

The model supports 13 distinct states:

- Susceptible: Individuals who can become infected
- Exposed: Infected but not yet infectious (incubation period)
- Prodromal: Infectious individuals in the community (replaces "Infected" in SEIR)
- Rash: Non-infectious individuals with visible symptoms (detection occurs here)
- Isolated: Detected individuals in self-isolation
- Isolated Recovered: Recovered individuals still in isolation
- Detected Hospitalized: Hospitalized individuals who were contact-traced
- Quarantined Exposed: Exposed individuals in quarantine due to contact tracing
- Quarantined Susceptible: Susceptible individuals in quarantine due to contact tracing
- Quarantined Prodromal: Prodromal individuals in quarantine due to contact tracing
- Quarantined Recovered: Recovered individuals in quarantine due to contact tracing
- Hospitalized: Individuals requiring hospital care
- Recovered: Individuals who have recovered and gained immunity

Template Parameters

<i>TSeq</i>	Type for genetic sequences (default: EPI_DEFAULT_TSEQ)
-------------	--

16.28.2 Constructor & Destructor Documentation

16.28.2.1 ModelMeaslesMixing() [1/2]

```
template<typename TSeq >
ModelMeaslesMixing< TSeq >::ModelMeaslesMixing (
    ModelMeaslesMixing< TSeq > & model,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double vax_efficacy,
    epiworld_double vax_reduction_recovery_rate,
    epiworld_double incubation_period,
    epiworld_double prodromal_period,
    epiworld_double rash_period,
    std::vector< double > contact_matrix,
    epiworld_double hospitalization_rate,
    epiworld_double hospitalization_period,
    epiworld_double days_undetected,
    epiworld_fast_int quarantine_period,
    epiworld_double quarantine_willingness,
    epiworld_double isolation_willingness,
    epiworld_fast_int isolation_period,
    epiworld_double prop_vaccinated,
    epiworld_double contact_tracing_success_rate = 1.0,
    epiworld_fast_uint contact_tracing_days_prior = 4u ) [inline]
```

Constructs a [ModelMeaslesMixing](#) object.

Template for a Measles model with population mixing, quarantine, and contact tracing.

Parameters

<i>model</i>	A reference to an existing ModelMeaslesMixing object.
<i>n</i>	The number of entities in the model.
<i>prevalence</i>	The initial prevalence of the disease in the model.
<i>contact_rate</i>	The contact rate between entities in the model.
<i>transmission_rate</i>	The transmission rate of the disease in the model.
<i>vax_efficacy</i>	The efficacy of the vaccine.
<i>vax_reduction_recovery_rate</i>	The reduction in recovery rate due to the vaccine.
<i>incubation_period</i>	The incubation period of the disease in the model.
<i>prodromal_period</i>	The prodromal period of the disease in the model.
<i>rash_period</i>	The rash period of the disease in the model.
<i>contact_matrix</i>	The contact matrix between entities in the model. Specified in column-major order.

Parameters

<i>hospitalization_rate</i>	The rate at which infected individuals are hospitalized.
<i>hospitalization_period</i>	The average duration of hospitalization in days.
<i>days_undetected</i>	The average number of days an infected individual remains undetected.
<i>quarantine_period</i>	The duration of quarantine in days for exposed contacts.
<i>quarantine_willingness</i>	The proportion of individuals willing to comply with quarantine measures.
<i>isolation_willingness</i>	The proportion of individuals willing to self-isolate when detected.
<i>isolation_period</i>	The duration of isolation in days for detected infected individuals.
<i>prop_vaccinated</i>	The proportion of vaccinated agents.
<i>contact_tracing_success_rate</i>	The probability of successfully identifying and tracing contacts (default: 1.0).
<i>contact_tracing_days_prior</i>	The number of days prior to detection for which contacts are traced (default: 4).
<i>model</i>	A ModelMeaslesMixing<TSeq> object where to set up the model.
<i>n</i>	Number of agents in the population
<i>prevalence</i>	Initial prevalence (proportion of infected individuals)
<i>contact_rate</i>	Average number of contacts (interactions) per step
<i>transmission_rate</i>	Probability of transmission per contact
<i>vax_efficacy</i>	The efficacy of the vaccine
<i>vax_reduction_recovery_rate</i>	The reduction in recovery rate due to the vaccine
<i>incubation_period</i>	Average incubation period in days
<i>prodromal_period</i>	Average prodromal period in days
<i>rash_period</i>	Average rash period in days
<i>contact_matrix</i>	Contact matrix specifying mixing patterns between population groups
<i>hospitalization_rate</i>	Rate at which infected individuals are hospitalized
<i>hospitalization_period</i>	Average duration of hospitalization in days
<i>days_undetected</i>	Average number of days an infected individual remains undetected
<i>quarantine_period</i>	Duration of quarantine in days for exposed contacts
<i>quarantine_willingness</i>	Proportion of individuals willing to comply with quarantine
<i>isolation_willingness</i>	Proportion of individuals willing to self-isolate when detected
<i>isolation_period</i>	Duration of isolation in days for detected infected individuals
<i>prop_vaccinated</i>	Proportion of vaccinated agents
<i>contact_tracing_success_rate</i>	Probability of successfully identifying contacts during tracing
<i>contact_tracing_days_prior</i>	Number of days prior to detection for contact tracing

16.28.2.2 ModelMeaslesMixing() [2/2]

```

template<typename TSeq >
ModelMeaslesMixing< TSeq >::ModelMeaslesMixing (
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double vax_efficacy,
    epiworld_double vax_reduction_recovery_rate,
    epiworld_double incubation_period,
    epiworld_double prodromal_period,

```

```

epiworld_double rash_period,
std::vector< double > contact_matrix,
epiworld_double hospitalization_rate,
epiworld_double hospitalization_period,
epiworld_double days_undetected,
epiworld_fast_int quarantine_period,
epiworld_double quarantine_willingness,
epiworld_double isolation_willingness,
epiworld_fast_int isolation_period,
epiworld_double prop_vaccinated,
epiworld_double contact_tracing_success_rate = 1.0,
epiworld_fast_uint contact_tracing_days_prior = 4u ) [inline]

```

Constructs a [ModelMeaslesMixing](#) object.

Parameters

<i>n</i>	The number of entities in the model.
<i>prevalence</i>	The initial prevalence of the disease in the model.
<i>contact_rate</i>	The contact rate between entities in the model.
<i>transmission_rate</i>	The transmission rate of the disease in the model.
<i>vax_efficacy</i>	The efficacy of the vaccine.
<i>vax_reduction_recovery_rate</i>	The reduction in recovery rate due to the vaccine.
<i>incubation_period</i>	The incubation period of the disease in the model.
<i>prodromal_period</i>	The prodromal period of the disease in the model.
<i>rash_period</i>	The rash period of the disease in the model.
<i>contact_matrix</i>	The contact matrix between entities in the model.
<i>hospitalization_rate</i>	The rate at which infected individuals are hospitalized.
<i>hospitalization_period</i>	The average duration of hospitalization in days.
<i>days_undetected</i>	The average number of days an infected individual remains undetected.
<i>quarantine_period</i>	The duration of quarantine in days for exposed contacts.
<i>quarantine_willingness</i>	The proportion of individuals willing to comply with quarantine measures.
<i>isolation_willingness</i>	The proportion of individuals willing to self-isolate when detected.
<i>isolation_period</i>	The duration of isolation in days for detected infected individuals.
<i>prop_vaccinated</i>	The proportion of vaccinated agents.
<i>contact_tracing_success_rate</i>	The probability of successfully identifying and tracing contacts (default: 1.0).
<i>contact_tracing_days_prior</i>	The number of days prior to detection for which contacts are traced (default: 4).

16.28.3 Member Function Documentation

16.28.3.1 clone_ptr()

```

template<typename TSeq >
Model< TSeq > * ModelMeaslesMixing< TSeq >::clone_ptr [inline], [virtual]

```

Create a clone of this model.

Returns

Pointer to a new model instance with the same configuration

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.28.3.2 get_agent_quarantine_triggered()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
std::vector< size_t > ModelMeaslesMixing< TSeq >::get_agent_quarantine_triggered ( ) const
[inline]
```

Get the quarantine trigger status for all agents.

Returns

Vector indicating quarantine process status for each agent

16.28.3.3 get_contact_matrix()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
std::vector< double > ModelMeaslesMixing< TSeq >::get_contact_matrix ( ) const [inline]
```

Get the current contact matrix.

Returns

Vector representing the contact matrix

16.28.3.4 get_isolation_willingness()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
std::vector< bool > ModelMeaslesMixing< TSeq >::get_isolation_willingness ( ) const [inline]
```

Get the isolation willingness for all agents.

Returns

Vector of boolean values indicating each agent's willingness to self-isolate

16.28.3.5 get_quarantine_willingness()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
std::vector< bool > ModelMeaslesMixing< TSeq >::get_quarantine_willingness ( ) const [inline]
```

Get the quarantine willingness for all agents.

Returns

Vector of boolean values indicating each agent's willingness to quarantine

16.28.3.6 initial_states()

```
template<typename TSeq >
ModelMeaslesMixing< TSeq > & ModelMeaslesMixing< TSeq >::initial_states (
    std::vector< double > proportions_,
    std::vector< int > queue_ = {} ) [inline], [virtual]
```

Set the initial states of the model.

Parameters

<i>proportions_</i>	Double vector with two elements: <ul style="list-style-type: none"> [0]: The proportion of initially infected individuals who start in the exposed state. [1]: The proportion of initially non-infected individuals who have recovered (immune).
<i>queue_</i>	Optional vector for queuing specifications (default: empty).

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.28.3.7 run()

```
template<typename TSeq >
ModelMeaslesMixing< TSeq > & ModelMeaslesMixing< TSeq >::run (
    epiworld_fast_uint ndays,
    int seed = -1 ) [inline], [virtual]
```

Run the model simulation.

Parameters

<i>ndays</i>	Number of days to simulate
<i>seed</i>	Random seed for reproducibility (default: -1 for random seed)

Returns

Reference to this model instance

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.28.3.8 set_contact_matrix()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
void ModelMeaslesMixing< TSeq >::set_contact_matrix (
    std::vector< double > cmat ) [inline]
```

Set the contact matrix for population mixing.

Parameters

<i>cmat</i>	Contact matrix specifying interaction rates between groups
-------------	--

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

- [include/epiworld/models/measlesmixing.hpp](#)

16.29 epiworld::epimodels::ModelMeaslesSchool< TSeq > Class Template Reference

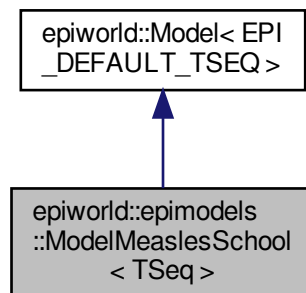
Template for a Measles model with quarantine.

```
#include <epiworld.hpp>
```

Inheritance diagram for epiworld::epimodels::ModelMeaslesSchool< TSeq >:



Collaboration diagram for `epiworld::epimodels::ModelMeaslesSchool< TSeq >`:



Public Member Functions

- void `quarantine_agents` ()
Quarantine agents that are in the system.
- void `reset` ()
Reset the model.
- void `update_infectious` ()
- `Model< TSeq > * clone_ptr` ()
Advanced usage: Makes a copy of data and returns it as undeleted pointer.
- `ModelMeaslesSchool` (`ModelMeaslesSchool< TSeq > &model`, `epiworld_fast_uint n`, `epiworld_fast_uint n_exposed`, `epiworld_double contact_rate`, `epiworld_double transmission_rate`, `epiworld_double vax_↵_efficacy`, `epiworld_double vax_reduction_recovery_rate`, `epiworld_double incubation_period`, `epiworld_↵_double prodromal_period`, `epiworld_double rash_period`, `epiworld_double days_undetected`, `epiworld_double hospitalization_rate`, `epiworld_double hospitalization_period`, `epiworld_double prop_vaccinated`, `epiworld_↵_fast_int quarantine_period`, `epiworld_double quarantine_willingness`, `epiworld_fast_int isolation_period`)
- **`ModelMeaslesSchool`** (`epiworld_fast_uint n`, `epiworld_fast_uint n_exposed`, `epiworld_double contact_↵_rate`, `epiworld_double transmission_rate`, `epiworld_double vax_efficacy`, `epiworld_double vax_reduction_↵_recovery_rate`, `epiworld_double incubation_period`, `epiworld_double prodromal_period`, `epiworld_double rash_period`, `epiworld_double days_undetected`, `epiworld_double hospitalization_rate`, `epiworld_double hospitalization_period`, `epiworld_double prop_vaccinated`, `epiworld_fast_int quarantine_period`, `epiworld_↵_double quarantine_willingness`, `epiworld_fast_int isolation_period`)

Public Attributes

- `std::vector< Agent< TSeq > * > infectious`
Agents infectious for contact.
- bool **`system_quarantine_triggered`** = false
- `std::vector< int > day_flagged`
Either detected or started quarantine.
- `std::vector< int > day_rash_onset`
Day of rash onset.

Static Public Attributes

- static const epiworld_fast_uint **SUSCEPTIBLE** = 0u
- static const epiworld_fast_uint **EXPOSED** = 1u
- static const epiworld_fast_uint **PRODROMAL** = 2u
- static const epiworld_fast_uint **RASH** = 3u
- static const epiworld_fast_uint **ISOLATED** = 4u
- static const epiworld_fast_uint **ISOLATED_RECOVERED** = 5u
- static const epiworld_fast_uint **DETECTED_HOSPITALIZED** = 6u
- static const epiworld_fast_uint **QUARANTINED_EXPOSED** = 7u
- static const epiworld_fast_uint **QUARANTINED_SUSCEPTIBLE** = 8u
- static const epiworld_fast_uint **QUARANTINED_PRODROMAL** = 9u
- static const epiworld_fast_uint **QUARANTINED_RECOVERED** = 10u
- static const epiworld_fast_uint **HOSPITALIZED** = 11u
- static const epiworld_fast_uint **RECOVERED** = 12u

Additional Inherited Members

16.29.1 Detailed Description

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
class epiworld::epimodels::ModelMeaslesSchool< TSeq >
```

Template for a Measles model with quarantine.

Parameters

<i>TSeq</i>	The type of the sequence to be used.
-------------	--------------------------------------

This model can be described as a SEIHR model with isolation and quarantine. The infectious state is divided into prodromal and rash phases. Furthermore, the quarantine state includes exposed, susceptible, prodromal, and recovered states.

The quarantine process is triggered any time that an agent with rash is detected. The agent is then isolated and all agents who are unvaccinated are quarantined. Isolated agents then may be moved out of the isolation in `isolation↔_period` days.

16.29.2 Constructor & Destructor Documentation

16.29.2.1 ModelMeaslesSchool()

```
template<typename TSeq >
ModelMeaslesSchool< TSeq >::ModelMeaslesSchool (
    ModelMeaslesSchool< TSeq > & model,
    epiworld_fast_uint n,
    epiworld_fast_uint n_exposed,
    epiworld_double contact_rate,
```

```

epiworld_double transmission_rate,
epiworld_double vax_efficacy,
epiworld_double vax_reduction_recovery_rate,
epiworld_double incubation_period,
epiworld_double prodromal_period,
epiworld_double rash_period,
epiworld_double days_undetected,
epiworld_double hospitalization_rate,
epiworld_double hospitalization_period,
epiworld_double prop_vaccinated,
epiworld_fast_int quarantine_period,
epiworld_double quarantine_willingness,
epiworld_fast_int isolation_period ) [inline]

```

Parameters

<i>n</i>	The number of agents in the system.
<i>n_exposed</i>	The number of exposed agents in the system.
<i>contact_rate</i>	The rate of contact between agents.
<i>transmission_rate</i>	The rate of transmission of the virus.
<i>vax_efficacy</i>	The efficacy of the vaccine.
<i>vax_reduction_recovery_rate</i>	The reduction in recovery rate due to the vaccine.
<i>incubation_period</i>	The incubation period of the virus.
<i>prodromal_period</i>	The prodromal period of the virus.
<i>rash_period</i>	The rash period of the virus.
<i>days_undetected</i>	The number of days the virus goes undetected.
<i>hospitalization_rate</i>	The rate of hospitalization.
<i>hospitalization_period</i>	The duration of hospitalization.
<i>prop_vaccinated</i>	The proportion of vaccinated agents.
<i>quarantine_period</i>	The number of days for quarantine.
<i>quarantine_willingness</i>	The willingness to be quarantined.
<i>isolation_period</i>	The number of days for isolation.

16.29.3 Member Function Documentation

16.29.3.1 clone_ptr()

```

template<typename TSeq >
Model< TSeq > * ModelMeaslesSchool< TSeq >::clone_ptr [inline], [virtual]

```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

<i>copy</i>	
-------------	--

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.29.3.2 quarantine_agents()

```
template<typename TSeq >
void ModelMeaslesSchool< TSeq >::quarantine_agents [inline]
```

Quarantine agents that are in the system.

The flow should be:

- The function only runs if the quarantine status is active.
- Agents who are in quarantine, isolation, removed, or hospitalized are ignored.
- Agents who are in the RASH state are isolated.
- Vaccinated agents are ignored.
- Susceptible, Exposed, and Prodromal agents are moved to the QUARANTINED_* state.
- At the end of the function, the quarantine status is set false.

16.29.3.3 reset()

```
template<typename TSeq >
void ModelMeaslesSchool< TSeq >::reset [inline], [virtual]
```

Reset the model.

Resetting the model will:

- clear the database
- restore the population (if `set_backup()` was called before)
- re-distribute tools
- re-distribute viruses
- set the date to 0

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

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

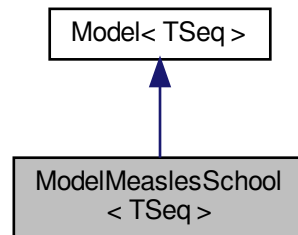
- `epiworld.hpp`

16.30 ModelMeaslesSchool< TSeq > Class Template Reference

Template for a Measles model with quarantine.

```
#include <measlesquarantine.hpp>
```

Inheritance diagram for ModelMeaslesSchool< TSeq >:



Collaboration diagram for ModelMeaslesSchool< TSeq >:



Public Member Functions

- void `quarantine_agents` ()
Quarantine agents that are in the system.
- void `reset` ()
Reset the model.
- void `update_infectious` ()

- [Model](#)< TSeq > * [clone_ptr](#) ()

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

- [ModelMeaslesSchool](#) ([ModelMeaslesSchool](#)< TSeq > &model, [epiworld_fast_uint](#) n, [epiworld_fast_uint](#) n_exposed, [epiworld_double](#) contact_rate, [epiworld_double](#) transmission_rate, [epiworld_double](#) vax_↵_efficacy, [epiworld_double](#) vax_reduction_recovery_rate, [epiworld_double](#) incubation_period, [epiworld_↵_double](#) prodromal_period, [epiworld_double](#) rash_period, [epiworld_double](#) days_undetected, [epiworld_double](#) hospitalization_rate, [epiworld_double](#) hospitalization_period, [epiworld_double](#) prop_vaccinated, [epiworld_↵_fast_int](#) quarantine_period, [epiworld_double](#) quarantine_willingness, [epiworld_fast_int](#) isolation_period)
- **ModelMeaslesSchool** ([epiworld_fast_uint](#) n, [epiworld_fast_uint](#) n_exposed, [epiworld_double](#) contact_↵_rate, [epiworld_double](#) transmission_rate, [epiworld_double](#) vax_efficacy, [epiworld_double](#) vax_reduction_↵_recovery_rate, [epiworld_double](#) incubation_period, [epiworld_double](#) prodromal_period, [epiworld_double](#) rash_period, [epiworld_double](#) days_undetected, [epiworld_double](#) hospitalization_rate, [epiworld_double](#) hospitalization_period, [epiworld_double](#) prop_vaccinated, [epiworld_fast_int](#) quarantine_period, [epiworld_↵_double](#) quarantine_willingness, [epiworld_fast_int](#) isolation_period)

Public Attributes

- [std::vector](#)< [Agent](#)< TSeq > * > [infectious](#)
Agents infectious for contact.
- [bool](#) **system_quarantine_triggered** = false
- [std::vector](#)< [int](#) > [day_flagged](#)
Either detected or started quarantine.
- [std::vector](#)< [int](#) > [day_rash_onset](#)
Day of rash onset.

Static Public Attributes

- [static const](#) [epiworld_fast_uint](#) **SUSCEPTIBLE** = 0u
- [static const](#) [epiworld_fast_uint](#) **EXPOSED** = 1u
- [static const](#) [epiworld_fast_uint](#) **PRODROMAL** = 2u
- [static const](#) [epiworld_fast_uint](#) **RASH** = 3u
- [static const](#) [epiworld_fast_uint](#) **ISOLATED** = 4u
- [static const](#) [epiworld_fast_uint](#) **ISOLATED_RECOVERED** = 5u
- [static const](#) [epiworld_fast_uint](#) **DETECTED_HOSPITALIZED** = 6u
- [static const](#) [epiworld_fast_uint](#) **QUARANTINED_EXPOSED** = 7u
- [static const](#) [epiworld_fast_uint](#) **QUARANTINED_SUSCEPTIBLE** = 8u
- [static const](#) [epiworld_fast_uint](#) **QUARANTINED_PRODROMAL** = 9u
- [static const](#) [epiworld_fast_uint](#) **QUARANTINED_RECOVERED** = 10u
- [static const](#) [epiworld_fast_uint](#) **HOSPITALIZED** = 11u
- [static const](#) [epiworld_fast_uint](#) **RECOVERED** = 12u

Additional Inherited Members

16.30.1 Detailed Description

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
class ModelMeaslesSchool< TSeq >
```

Template for a Measles model with quarantine.

Parameters

<i>TSeq</i>	The type of the sequence to be used.
-------------	--------------------------------------

This model can be described as a SEIHR model with isolation and quarantine. The infectious state is divided into prodromal and rash phases. Furthermore, the quarantine state includes exposed, susceptible, prodromal, and recovered states.

The quarantine process is triggered any time that an agent with rash is detected. The agent is then isolated and all agents who are unvaccinated are quarantined. Isolated agents then may be moved out of the isolation in `isolation_↵_period` days.

16.30.2 Constructor & Destructor Documentation

16.30.2.1 ModelMeaslesSchool()

```
template<typename TSeq >
ModelMeaslesSchool< TSeq >::ModelMeaslesSchool (
    ModelMeaslesSchool< TSeq > & model,
    epiworld_fast_uint n,
    epiworld_fast_uint n_exposed,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double vax_efficacy,
    epiworld_double vax_reduction_recovery_rate,
    epiworld_double incubation_period,
    epiworld_double prodromal_period,
    epiworld_double rash_period,
    epiworld_double days_undetected,
    epiworld_double hospitalization_rate,
    epiworld_double hospitalization_period,
    epiworld_double prop_vaccinated,
    epiworld_fast_int quarantine_period,
    epiworld_double quarantine_willingness,
    epiworld_fast_int isolation_period ) [inline]
```

Parameters

<i>n</i>	The number of agents in the system.
<i>n_exposed</i>	The number of exposed agents in the system.
<i>contact_rate</i>	The rate of contact between agents.
<i>transmission_rate</i>	The rate of transmission of the virus.
<i>vax_efficacy</i>	The efficacy of the vaccine.
<i>vax_reduction_recovery_rate</i>	The reduction in recovery rate due to the vaccine.
<i>incubation_period</i>	The incubation period of the virus.
<i>prodromal_period</i>	The prodromal period of the virus.
<i>rash_period</i>	The rash period of the virus.
<i>days_undetected</i>	The number of days the virus goes undetected.
<i>hospitalization_rate</i>	The rate of hospitalization.

Parameters

<i>hospitalization_period</i>	The duration of hospitalization.
<i>prop_vaccinated</i>	The proportion of vaccinated agents.
<i>quarantine_period</i>	The number of days for quarantine.
<i>quarantine_willingness</i>	The willingness to be quarantined.
<i>isolation_period</i>	The number of days for isolation.

16.30.3 Member Function Documentation

16.30.3.1 clone_ptr()

```
template<typename TSeq >
Model< TSeq > * ModelMeaslesSchool< TSeq >::clone_ptr [inline], [virtual]
```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

<i>copy</i>	
-------------	--

Reimplemented from [Model< TSeq >](#).

16.30.3.2 quarantine_agents()

```
template<typename TSeq >
void ModelMeaslesSchool< TSeq >::quarantine_agents [inline]
```

Quarantine agents that are in the system.

The flow should be:

- The function only runs if the quarantine status is active.
- Agents who are in quarantine, isolation, removed, or hospitalized are ignored.
- Agents who are in the RASH state are isolated.
- Vaccinated agents are ignored.
- Susceptible, Exposed, and Prodromal agents are moved to the QUARANTINED_* state.
- At the end of the function, the quarantine status is set false.

16.30.3.3 reset()

```
template<typename TSeq >
void ModelMeaslesSchool< TSeq >::reset [inline], [virtual]
```

Reset the model.

Resetting the model will:

- clear the database
- restore the population (if `set_backup()` was called before)
- re-distribute tools
- re-distribute viruses
- set the date to 0

Reimplemented from [Model< TSeq >](#).

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

- `include/epiworld/models/measlesquarantine.hpp`

16.31 epiworld::epimodels::ModelSEIR< TSeq > Class Template Reference

Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.

```
#include <epiworld.hpp>
```

Inheritance diagram for `epiworld::epimodels::ModelSEIR< TSeq >`:



Collaboration diagram for epiworld::epimodels::ModelSEIR< TSeq >:



Public Member Functions

- **ModelSEIR** ([ModelSEIR](#)< TSeq > &model, const std::string &vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate↔
rate)
- **ModelSEIR** (const std::string &vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate)
- [ModelSEIR](#)< TSeq > & [initial_states](#) (std::vector< double > proportions_, std::vector< int > queue_={})
Set up the initial states of the model.

Public Attributes

- epiworld::UpdateFun< TSeq > **update_exposed_seir**
- epiworld::UpdateFun< TSeq > **update_infected_seir**

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **EXPOSED** = 1
- static const int **INFECTED** = 2
- static const int **REMOVED** = 3

Additional Inherited Members

16.31.1 Detailed Description

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
class epiworld::epimodels::ModelSEIR< TSeq >
```

Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.

Parameters

<i>model</i>	A Model<TSeq> object where to set up the SIR.
<i>vname</i>	std::string Name of the virus
<i>prevalence</i>	epiworld_double Initial prevalence the immune system
<i>transmission_rate</i>	epiworld_double Transmission rate of the virus
<i>avg_incubation_days</i>	epiworld_double Average incubation days of the virus.
<i>recovery_rate</i>	epiworld_double Recovery rate of the virus.

16.31.2 Member Function Documentation

16.31.2.1 initial_states()

```
template<typename TSeq >
ModelSEIR< TSeq > & ModelSEIR< TSeq >::initial_states (
    std::vector< double > proportions_,
    std::vector< int > queue_ = {} ) [inline], [virtual]
```

Set up the initial states of the model.

Parameters

<i>proportions_</i> ↔	Double vector with the following values:
—	<ul style="list-style-type: none"> • 0: Proportion of non-infected agents who are removed. • 1: Proportion of exposed agents to be set as infected.

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.31.3 Member Data Documentation

16.31.3.1 update_exposed_seir

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
epiworld::UpdateFun<TSeq> epiworld::epimodels::ModelSEIR< TSeq >::update_exposed_seir
```

Initial value:

```
= [] (
    epiworld::Agent<TSeq> * p,
    epiworld::Model<TSeq> * m
) -> void {
    auto v = p->get_virus();
    if (m->runif() < 1.0/(v->get_incubation(m)))
        p->change_state(m, ModelSEIR<TSeq>::INFECTED);
    return;
}
```

16.31.3.2 update_infected_seir

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
epiworld::UpdateFun<TSeq> epiworld::epimodels::ModelSEIR< TSeq >::update_infected_seir
```

Initial value:

```
= [] (
    epiworld::Agent<TSeq> * p,
    epiworld::Model<TSeq> * m
) -> void {
    if (m->runif() < (m->par("Recovery rate")))
        p->rm_virus(m);
    return;
}
```

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

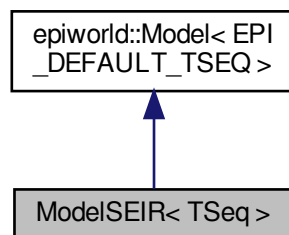
- epiworld.hpp

16.32 ModelSEIR< TSeq > Class Template Reference

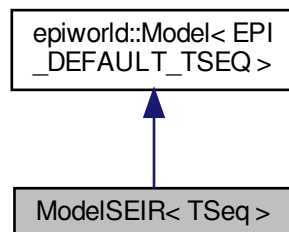
Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.

```
#include <seir.hpp>
```

Inheritance diagram for ModelSEIR< TSeq >:



Collaboration diagram for ModelSEIR< TSeq >:



Public Member Functions

- **ModelSEIR** ([ModelSEIR](#)< TSeq > &model, const std::string &vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate↵rate)
- **ModelSEIR** (const std::string &vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate)
- [ModelSEIR](#)< TSeq > & [initial_states](#) (std::vector< double > proportions_, std::vector< int > queue_={})
Set up the initial states of the model.

Public Attributes

- epiworld::UpdateFun< TSeq > **update_exposed_seir**
- epiworld::UpdateFun< TSeq > **update_infected_seir**

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **EXPOSED** = 1
- static const int **INFECTED** = 2
- static const int **REMOVED** = 3

Additional Inherited Members

16.32.1 Detailed Description

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
class ModelSEIR< TSeq >
```

Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.

Parameters

<i>model</i>	A Model<TSeq> object where to set up the SIR.
<i>vname</i>	std::string Name of the virus
<i>prevalence</i>	epiworld_double Initial prevalence the immune system
<i>transmission_rate</i>	epiworld_double Transmission rate of the virus
<i>avg_incubation_days</i>	epiworld_double Average incubation days of the virus.
<i>recovery_rate</i>	epiworld_double Recovery rate of the virus.

16.32.2 Member Function Documentation

16.32.2.1 initial_states()

```
template<typename TSeq >
ModelSEIR< TSeq > & ModelSEIR< TSeq >::initial_states (
    std::vector< double > proportions_,
    std::vector< int > queue_ = {} ) [inline], [virtual]
```

Set up the initial states of the model.

Parameters

<i>proportions_</i> ↔	Double vector with the following values:
—	<ul style="list-style-type: none"> • 0: Proportion of non-infected agents who are removed. • 1: Proportion of exposed agents to be set as infected.

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.32.3 Member Data Documentation

16.32.3.1 update_exposed_seir

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
epiworld::UpdateFun<TSeq> ModelSEIR< TSeq >::update_exposed_seir
```

Initial value:

```
= [] (
    epiworld::Agent<TSeq> * p,
    epiworld::Model<TSeq> * m
) -> void {
    auto v = p->get_virus();
    if (m->runif() < 1.0/(v->get_incubation(m)))
        p->change_state(m, ModelSEIR<TSeq>::INFECTED);
    return;
}
```

16.32.3.2 update_infected_seir

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
epiworld::UpdateFun<TSeq> ModelSEIR< TSeq >::update_infected_seir
```

Initial value:

```
= [] (
    epiworld::Agent<TSeq> * p,
    epiworld::Model<TSeq> * m
) -> void {
    if (m->runif() < (m->par("Recovery rate")))
        p->rm_virus(m);
    return;
}
```

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

- include/epiworld/models/seir.hpp

16.33 epiworld::epimodels::ModelSEIRCONN< TSeq > Class Template Reference

Inheritance diagram for epiworld::epimodels::ModelSEIRCONN< TSeq >:



Collaboration diagram for epiworld::epimodels::ModelSEIRCONN< TSeq >:



Public Member Functions

- [ModelSEIRCONN](#) ([ModelSEIRCONN](#)< TSeq > &model, const std::string &vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate)
Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.
- **ModelSEIRCONN** (const std::string &vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate)
- [ModelSEIRCONN](#)< TSeq > & [run](#) (epiworld_fast_uint ndays, int seed=-1)
Runs the simulation (after initialization)
- void [reset](#) ()
Reset the model.

- `Model< TSeq > * clone_ptr ()`
Advanced usage: Makes a copy of data and returns it as undeleted pointer.
- `ModelSEIRCONN< TSeq > & initial_states (std::vector< double > proportions_, std::vector< int > queue←_={})`
Set the initial states of the model.
- `size_t get_n_infected () const`
- `std::vector< double > generation_time_expected (int max_days=200, int max_contacts=200) const`

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **EXPOSED** = 1
- static const int **INFECTED** = 2
- static const int **RECOVERED** = 3

Additional Inherited Members

16.33.1 Constructor & Destructor Documentation

16.33.1.1 ModelSEIRCONN()

```
template<typename TSeq >
ModelSEIRCONN< TSeq >::ModelSEIRCONN (
    ModelSEIRCONN< TSeq > & model,
    const std::string & vname,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double avg_incubation_days,
    epiworld_double recovery_rate ) [inline]
```

Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.

Parameters

<i>model</i>	A Model<TSeq> object where to set up the SIR.
<i>vname</i>	std::string Name of the virus
<i>prevalence</i>	Initial prevalence (proportion)
<i>contact_rate</i>	Average number of contacts (interactions) per step.
<i>transmission_rate</i>	Probability of transmission
<i>recovery_rate</i>	Probability of recovery

16.33.2 Member Function Documentation

16.33.2.1 clone_ptr()

```
template<typename TSeq >
Model< TSeq > * ModelSEIRCONN< TSeq >::clone_ptr [inline], [virtual]
```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

<i>copy</i>	
-------------	--

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.33.2.2 initial_states()

```
template<typename TSeq >
ModelSEIRCONN< TSeq > & ModelSEIRCONN< TSeq >::initial_states (
    std::vector< double > proportions_,
    std::vector< int > queue_ = {} ) [inline], [virtual]
```

Set the initial states of the model.

Parameters

<i>proportions_</i> ↔	Double vector with a single element:
—	<ul style="list-style-type: none"> The proportion of non-infected individuals who have recovered.

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.33.2.3 reset()

```
template<typename TSeq >
void ModelSEIRCONN< TSeq >::reset [inline], [virtual]
```

Reset the model.

Resetting the model will:

- clear the database
- restore the population (if `set_backup()` was called before)
- re-distribute tools
- re-distribute viruses
- set the date to 0

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

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

- `epiworld.hpp`

16.34 ModelSEIRCONN< TSeq > Class Template Reference

Inheritance diagram for ModelSEIRCONN< TSeq >:



Collaboration diagram for ModelSEIRCONN< TSeq >:



Public Member Functions

- [ModelSEIRCONN](#) ([ModelSEIRCONN](#)< TSeq > &model, const std::string &vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate)
Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.
- **ModelSEIRCONN** (const std::string &vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate)
- [ModelSEIRCONN](#)< TSeq > & [run](#) (epiworld_fast_uint ndays, int seed=-1)
Runs the simulation (after initialization)
- void [reset](#) ()
Reset the model.
- [Model](#)< TSeq > * [clone_ptr](#) ()
Advanced usage: Makes a copy of data and returns it as undeleted pointer.

- `ModelSEIRCONN< TSeq > & initial_states` (`std::vector< double > proportions_`, `std::vector< int > queue_` ← `_={}`)
Set the initial states of the model.
- `size_t get_n_infected ()` const
- `std::vector< double > generation_time_expected` (`int max_days=200`, `int max_contacts=200`) const

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **EXPOSED** = 1
- static const int **INFECTED** = 2
- static const int **RECOVERED** = 3

Additional Inherited Members

16.34.1 Constructor & Destructor Documentation

16.34.1.1 ModelSEIRCONN()

```
template<typename TSeq >
ModelSEIRCONN< TSeq >::ModelSEIRCONN (
    ModelSEIRCONN< TSeq > & model,
    const std::string & vname,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double avg_incubation_days,
    epiworld_double recovery_rate ) [inline]
```

Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.

Parameters

<i>model</i>	A Model<TSeq> object where to set up the SIR.
<i>vname</i>	std::string Name of the virus
<i>prevalence</i>	Initial prevalence (proportion)
<i>contact_rate</i>	Average number of contacts (interactions) per step.
<i>transmission_rate</i>	Probability of transmission
<i>recovery_rate</i>	Probability of recovery

16.34.2 Member Function Documentation

16.34.2.1 clone_ptr()

```
template<typename TSeq >
Model< TSeq > * ModelSEIRCONN< TSeq >::clone_ptr [inline], [virtual]
```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

<i>copy</i>	
-------------	--

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.34.2.2 initial_states()

```
template<typename TSeq >
ModelSEIRCONN< TSeq > & ModelSEIRCONN< TSeq >::initial_states (
    std::vector< double > proportions_,
    std::vector< int > queue_ = {} ) [inline], [virtual]
```

Set the initial states of the model.

Parameters

<i>proportions_</i> ↔	Double vector with a single element:
—	<ul style="list-style-type: none"> The proportion of non-infected individuals who have recovered.

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.34.2.3 reset()

```
template<typename TSeq >
void ModelSEIRCONN< TSeq >::reset [inline], [virtual]
```

Reset the model.

Resetting the model will:

- clear the database
- restore the population (if `set_backup()` was called before)
- re-distribute tools
- re-distribute viruses
- set the date to 0

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

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

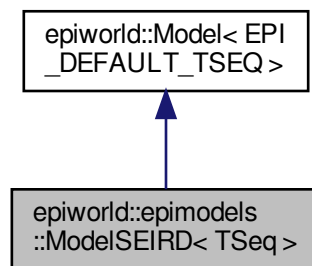
- `include/epiworld/models/seirconnected.hpp`

16.35 epiworld::epimodels::ModelSEIRD< TSeq > Class Template Reference

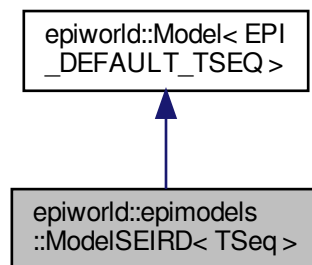
Template for a Susceptible-Exposed-Infected-Removed-Deceased (SEIRD) model.

```
#include <epiworld.hpp>
```

Inheritance diagram for epiworld::epimodels::ModelSEIRD< TSeq >:



Collaboration diagram for epiworld::epimodels::ModelSEIRD< TSeq >:



Public Member Functions

- [ModelSEIRD](#) ([ModelSEIRD](#)< TSeq > &model, const std::string &vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate, epiworld_double death_rate)
Constructor for the SEIRD model.
- [ModelSEIRD](#) (const std::string &vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate, epiworld_double death_rate)
Constructor for the SEIRD model.
- [ModelSEIRD](#)< TSeq > & **initial_states** (std::vector< double > proportions_, std::vector< int > queue_={})

Public Attributes

- epiworld::UpdateFun< TSeq > **update_exposed_seir**
- epiworld::UpdateFun< TSeq > **update_infected**

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **EXPOSED** = 1
- static const int **INFECTED** = 2
- static const int **REMOVED** = 3
- static const int **DECEASED** = 4

Additional Inherited Members

16.35.1 Detailed Description

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
class epiworld::epimodels::ModelSEIRD< TSeq >
```

Template for a Susceptible-Exposed-Infected-Removed-Deceased (SEIRD) model.

16.35.2 Constructor & Destructor Documentation

16.35.2.1 ModelSEIRD() [1/2]

```
template<typename TSeq >
ModelSEIRD< TSeq >::ModelSEIRD (
    ModelSEIRD< TSeq > & model,
    const std::string & vname,
    epiworld_double prevalence,
    epiworld_double transmission_rate,
    epiworld_double avg_incubation_days,
    epiworld_double recovery_rate,
    epiworld_double death_rate ) [inline]
```

Constructor for the SEIRD model.

Template Parameters

<i>TSeq</i>	Type of the sequence used in the model.
-------------	---

Parameters

<i>model</i>	Reference to the SEIRD model.
<i>vname</i>	Name of the model.

Parameters

<i>prevalence</i>	Prevalence of the disease.
<i>transmission_rate</i>	Transmission rate of the disease.
<i>avg_incubation_days</i>	Average incubation period of the disease.
<i>recovery_rate</i>	Recovery rate of the disease.
<i>death_rate</i>	Death rate of the disease.

16.35.2.2 ModelSEIRD() [2/2]

```
template<typename TSeq >
ModelSEIRD< TSeq >::ModelSEIRD (
    const std::string & vname,
    epiworld_double prevalence,
    epiworld_double transmission_rate,
    epiworld_double avg_incubation_days,
    epiworld_double recovery_rate,
    epiworld_double death_rate ) [inline]
```

Constructor for the SEIRD model.

Parameters

<i>vname</i>	Name of the model.
<i>prevalence</i>	Initial prevalence of the disease.
<i>transmission_rate</i>	Transmission rate of the disease.
<i>avg_incubation_days</i>	Average incubation period of the disease.
<i>recovery_rate</i>	Recovery rate of the disease.
<i>death_rate</i>	Death rate of the disease.

16.35.3 Member Data Documentation**16.35.3.1 update_exposed_seir**

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
epiworld::UpdateFun<TSeq> epiworld::epimodels::ModelSEIRD< TSeq >::update_exposed_seir
```

Initial value:

```
= [] (
    epiworld::Agent<TSeq> * p,
    epiworld::Model<TSeq> * m
) -> void {
    auto v = p->get_virus();
    if (m->runif() < 1.0/(v->get_incubation(m)))
        p->change_state(m, ModelSEIRD<TSeq>::INFECTED);
    return;
}
```

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

- `epiworld.hpp`

16.36 ModelSEIRD< TSeq > Class Template Reference

Template for a Susceptible-Exposed-Infected-Removed-Deceased (SEIRD) model.

```
#include <seird.hpp>
```

Inheritance diagram for ModelSEIRD< TSeq >:



Collaboration diagram for ModelSEIRD< TSeq >:



Public Member Functions

- **ModelSEIRD** (**ModelSEIRD**< TSeq > &model, const std::string &vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate, epiworld_double death_rate)
Constructor for the SEIRD model.
- **ModelSEIRD** (const std::string &vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate, epiworld_double death_rate)
Constructor for the SEIRD model.
- **ModelSEIRD**< TSeq > & **initial_states** (std::vector< double > proportions_, std::vector< int > queue_={})

Public Attributes

- `epiworld::UpdateFun< TSeq > update_exposed_seir`
- `epiworld::UpdateFun< TSeq > update_infected`

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **EXPOSED** = 1
- static const int **INFECTED** = 2
- static const int **REMOVED** = 3
- static const int **DECEASED** = 4

Additional Inherited Members

16.36.1 Detailed Description

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
class ModelSEIRD< TSeq >
```

Template for a Susceptible-Exposed-Infected-Removed-Deceased (SEIRD) model.

16.36.2 Constructor & Destructor Documentation

16.36.2.1 ModelSEIRD() [1/2]

```
template<typename TSeq >
ModelSEIRD< TSeq >::ModelSEIRD (
    ModelSEIRD< TSeq > & model,
    const std::string & vname,
    epiworld_double prevalence,
    epiworld_double transmission_rate,
    epiworld_double avg_incubation_days,
    epiworld_double recovery_rate,
    epiworld_double death_rate ) [inline]
```

Constructor for the SEIRD model.

Template Parameters

<i>TSeq</i>	Type of the sequence used in the model.
-------------	---

Parameters

<i>model</i>	Reference to the SEIRD model.
<i>vname</i>	Name of the model.

Parameters

<i>prevalence</i>	Prevalence of the disease.
<i>transmission_rate</i>	Transmission rate of the disease.
<i>avg_incubation_days</i>	Average incubation period of the disease.
<i>recovery_rate</i>	Recovery rate of the disease.
<i>death_rate</i>	Death rate of the disease.

16.36.2.2 ModelSEIRD() [2/2]

```
template<typename TSeq >
ModelSEIRD< TSeq >::ModelSEIRD (
    const std::string & vname,
    epiworld_double prevalence,
    epiworld_double transmission_rate,
    epiworld_double avg_incubation_days,
    epiworld_double recovery_rate,
    epiworld_double death_rate ) [inline]
```

Constructor for the SEIRD model.

Parameters

<i>vname</i>	Name of the model.
<i>prevalence</i>	Initial prevalence of the disease.
<i>transmission_rate</i>	Transmission rate of the disease.
<i>avg_incubation_days</i>	Average incubation period of the disease.
<i>recovery_rate</i>	Recovery rate of the disease.
<i>death_rate</i>	Death rate of the disease.

16.36.3 Member Data Documentation

16.36.3.1 update_exposed_seir

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
epiworld::UpdateFun<TSeq> ModelSEIRD< TSeq >::update_exposed_seir
```

Initial value:

```
= [] (
    epiworld::Agent<TSeq> * p,
    epiworld::Model<TSeq> * m
) -> void {
    auto v = p->get_virus();
    if (m->runif() < 1.0/(v->get_incubation(m)))
        p->change_state(m, ModelSEIRD<TSeq>::INFECTED);
    return;
}
```

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

- `include/epiworld/models/seird.hpp`

16.37 `epiworld::epimodels::ModelSEIRDCONN< TSeq >` Class Template Reference

Inheritance diagram for `epiworld::epimodels::ModelSEIRDCONN< TSeq >`:



Collaboration diagram for `epiworld::epimodels::ModelSEIRDCONN< TSeq >`:



Public Member Functions

- [ModelSEIRDCONN](#) ([ModelSEIRDCONN](#)< TSeq > &model, const std::string &vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate, epiworld_double death_rate)
Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.
- **ModelSEIRDCONN** (const std::string &vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate, epiworld_double death_rate)
- [ModelSEIRDCONN](#)< TSeq > & [run](#) (epiworld_fast_uint ndays, int seed=-1)

- Runs the simulation (after initialization)*
- void `reset` ()
Reset the model.
- `Model< TSeq > * clone_ptr` ()
Advanced usage: Makes a copy of data and returns it as undeleted pointer.
- `ModelSEIRDCONN< TSeq > & initial_states` (std::vector< double > proportions_, std::vector< int > queue_={})
Set up the initial states of the model.
- size_t `get_n_infected` () const

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **EXPOSED** = 1
- static const int **INFECTED** = 2
- static const int **REMOVED** = 3
- static const int **DECEASED** = 4

Additional Inherited Members

16.37.1 Constructor & Destructor Documentation

16.37.1.1 ModelSEIRDCONN()

```
template<typename TSeq >
ModelSEIRDCONN< TSeq >::ModelSEIRDCONN (
    ModelSEIRDCONN< TSeq > & model,
    const std::string & vname,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double avg_incubation_days,
    epiworld_double recovery_rate,
    epiworld_double death_rate ) [inline]
```

Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.

Parameters

<i>model</i>	A Model<TSeq> object where to set up the SIR.
<i>vname</i>	std::string Name of the virus
<i>prevalence</i>	Initial prevalence (proportion)
<i>contact_rate</i>	Average number of contacts (interactions) per step.
<i>transmission_rate</i>	Probability of transmission
<i>recovery_rate</i>	Probability of recovery
<i>death_rate</i>	Probability of death

16.37.2 Member Function Documentation

16.37.2.1 clone_ptr()

```
template<typename TSeq >
Model< TSeq > * ModelSEIRDCONN< TSeq >::clone_ptr [inline], [virtual]
```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

<i>copy</i>	
-------------	--

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.37.2.2 initial_states()

```
template<typename TSeq >
ModelSEIRDCONN< TSeq > & ModelSEIRDCONN< TSeq >::initial_states (
    std::vector< double > proportions_,
    std::vector< int > queue_ = {} ) [inline], [virtual]
```

Set up the initial states of the model.

Parameters

<i>proportions_↵</i>	Double vector with the following values:
—	<ul style="list-style-type: none"> • 0: Proportion of non-infected agents who are removed. • 1: Proportion of exposed agents to be set as infected.

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.37.2.3 reset()

```
template<typename TSeq >
void ModelSEIRDCONN< TSeq >::reset [inline], [virtual]
```

Reset the model.

Resetting the model will:

- clear the database
- restore the population (if `set_backup()` was called before)
- re-distribute tools
- re-distribute viruses
- set the date to 0

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

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

- `epiworld.hpp`

16.38 ModelSEIRDCONN< TSeq > Class Template Reference

Inheritance diagram for ModelSEIRDCONN< TSeq >:



Collaboration diagram for ModelSEIRDCONN< TSeq >:



Public Member Functions

- [ModelSEIRDCONN](#) ([ModelSEIRDCONN](#)< TSeq > &model, const std::string &vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate, epiworld_double death_rate)
Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.
- **ModelSEIRDCONN** (const std::string &vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate, epiworld_double death_rate)
- [ModelSEIRDCONN](#)< TSeq > & [run](#) (epiworld_fast_uint ndays, int seed=-1)
Runs the simulation (after initialization)
- void [reset](#) ()
Reset the model.
- [Model](#)< TSeq > * [clone_ptr](#) ()
Advanced usage: Makes a copy of data and returns it as undeleted pointer.
- [ModelSEIRDCONN](#)< TSeq > & [initial_states](#) (std::vector< double > proportions_, std::vector< int > queue_={})
Set up the initial states of the model.
- size_t [get_n_infected](#) () const

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **EXPOSED** = 1
- static const int **INFECTED** = 2
- static const int **REMOVED** = 3
- static const int **DECEASED** = 4

Additional Inherited Members

16.38.1 Constructor & Destructor Documentation

16.38.1.1 ModelSEIRDCONN()

```
template<typename TSeq >
ModelSEIRDCONN< TSeq >::ModelSEIRDCONN (
    ModelSEIRDCONN< TSeq > & model,
    const std::string & vname,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double avg_incubation_days,
    epiworld_double recovery_rate,
    epiworld_double death_rate ) [inline]
```

Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.

Parameters

<i>model</i>	A Model<TSeq> object where to set up the SIR.
<i>vname</i>	std::string Name of the virus
<i>prevalence</i>	Initial prevalence (proportion)
<i>contact_rate</i>	Average number of contacts (interactions) per step.
<i>transmission_rate</i>	Probability of transmission
<i>recovery_rate</i>	Probability of recovery
<i>death_rate</i>	Probability of death

16.38.2 Member Function Documentation

16.38.2.1 clone_ptr()

```
template<typename TSeq >
Model< TSeq > * ModelSEIRDCONN< TSeq >::clone_ptr [inline], [virtual]
```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

<i>copy</i>	
-------------	--

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.38.2.2 initial_states()

```
template<typename TSeq >
ModelSEIRDCONN< TSeq > & ModelSEIRDCONN< TSeq >::initial_states (
    std::vector< double > proportions_,
    std::vector< int > queue_ = {} ) [inline], [virtual]
```

Set up the initial states of the model.

Parameters

<i>proportions_↵</i>	Double vector with the following values:
—	<ul style="list-style-type: none"> • 0: Proportion of non-infected agents who are removed. • 1: Proportion of exposed agents to be set as infected.

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.38.2.3 reset()

```
template<typename TSeq >
void ModelSEIRDCONN< TSeq >::reset [inline], [virtual]
```

Reset the model.

Resetting the model will:

- clear the database
- restore the population (if `set_backup()` was called before)
- re-distribute tools
- re-distribute viruses
- set the date to 0

Reimplemented from `epiworld::Model< EPI_DEFAULT_TSEQ >`.

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

- `include/epiworld/models/seirdconnected.hpp`

16.39 epiworld::epimodels::ModelSEIRMixing< TSeq > Class Template Reference

Inheritance diagram for `epiworld::epimodels::ModelSEIRMixing< TSeq >`:



Collaboration diagram for epiworld::epimodels::ModelSEIRMixing< TSeq >:



Public Member Functions

- [ModelSEIRMixing](#) ([ModelSEIRMixing](#)< TSeq > &model, const std::string &vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate, std::vector< double > contact_matrix)
Constructs a [ModelSEIRMixing](#) object.
- [ModelSEIRMixing](#) (const std::string &vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate, std::vector< double > contact_matrix)
Constructs a [ModelSEIRMixing](#) object.
- [ModelSEIRMixing](#)< TSeq > & [run](#) (epiworld_fast_uint ndays, int seed=-1)
Runs the simulation (after initialization)
- void [reset](#) ()
Reset the model.
- [Model](#)< TSeq > * [clone_ptr](#) ()
Advanced usage: Makes a copy of data and returns it as undeleted pointer.
- [ModelSEIRMixing](#)< TSeq > & [initial_states](#) (std::vector< double > proportions_, std::vector< int > queue_ = {})
Set the initial states of the model.
- void [set_contact_matrix](#) (std::vector< double > cmat)

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **EXPOSED** = 1
- static const int **INFECTED** = 2
- static const int **RECOVERED** = 3

Additional Inherited Members

16.39.1 Constructor & Destructor Documentation

16.39.1.1 ModelSEIRMixing() [1/2]

```
template<typename TSeq >
ModelSEIRMixing< TSeq >::ModelSEIRMixing (
    ModelSEIRMixing< TSeq > & model,
    const std::string & vname,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double avg_incubation_days,
    epiworld_double recovery_rate,
    std::vector< double > contact_matrix ) [inline]
```

Constructs a [ModelSEIRMixing](#) object.

Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.

Parameters

<i>model</i>	A reference to an existing ModelSEIRMixing object.
<i>vname</i>	The name of the ModelSEIRMixing object.
<i>n</i>	The number of entities in the model.
<i>prevalence</i>	The initial prevalence of the disease in the model.
<i>contact_rate</i>	The contact rate between entities in the model.
<i>transmission_rate</i>	The transmission rate of the disease in the model.
<i>avg_incubation_days</i>	The average incubation period of the disease in the model.
<i>recovery_rate</i>	The recovery rate of the disease in the model.
<i>contact_matrix</i>	The contact matrix between entities in the model. Specified in column-major order.
<i>model</i>	A Model<TSeq> object where to set up the SIR.
<i>vname</i>	std::string Name of the virus
<i>prevalence</i>	Initial prevalence (proportion)
<i>contact_rate</i>	Average number of contacts (interactions) per step.
<i>transmission_rate</i>	Probability of transmission
<i>recovery_rate</i>	Probability of recovery

16.39.1.2 ModelSEIRMixing() [2/2]

```
template<typename TSeq >
ModelSEIRMixing< TSeq >::ModelSEIRMixing (
    const std::string & vname,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double avg_incubation_days,
    epiworld_double recovery_rate,
    std::vector< double > contact_matrix ) [inline]
```

Constructs a [ModelSEIRMixing](#) object.

Parameters

<i>vname</i>	The name of the ModelSEIRMixing object.
<i>n</i>	The number of entities in the model.
<i>prevalence</i>	The initial prevalence of the disease in the model.
<i>contact_rate</i>	The contact rate between entities in the model.
<i>transmission_rate</i>	The transmission rate of the disease in the model.
<i>avg_incubation_days</i>	The average incubation period of the disease in the model.
<i>recovery_rate</i>	The recovery rate of the disease in the model.
<i>contact_matrix</i>	The contact matrix between entities in the model.

16.39.2 Member Function Documentation

16.39.2.1 `clone_ptr()`

```
template<typename TSeq >
Model< TSeq > * ModelSEIRMixing< TSeq >::clone_ptr [inline], [virtual]
```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

<i>copy</i>	
-------------	--

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.39.2.2 `initial_states()`

```
template<typename TSeq >
ModelSEIRMixing< TSeq > & ModelSEIRMixing< TSeq >::initial_states (
    std::vector< double > proportions_,
    std::vector< int > queue_ = {} ) [inline], [virtual]
```

Set the initial states of the model.

Parameters

<i>proportions_↵</i>	Double vector with a single element:
—	<ul style="list-style-type: none"> The proportion of non-infected individuals who have recovered.

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.39.2.3 reset()

```
template<typename TSeq >
void ModelSEIRMixing< TSeq >::reset [inline], [virtual]
```

Reset the model.

Resetting the model will:

- clear the database
- restore the population (if `set_backup()` was called before)
- re-distribute tools
- re-distribute viruses
- set the date to 0

Reimplemented from `epiworld::Model< EPI_DEFAULT_TSEQ >`.

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

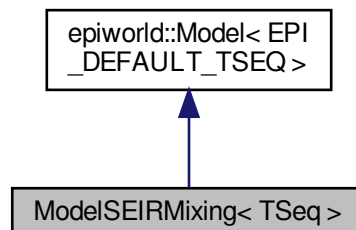
- `epiworld.hpp`

16.40 ModelSEIRMixing< TSeq > Class Template Reference

Inheritance diagram for ModelSEIRMixing< TSeq >:



Collaboration diagram for ModelSEIRMixing< TSeq >:



Public Member Functions

- [ModelSEIRMixing](#) ([ModelSEIRMixing](#)< TSeq > &model, const std::string &vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate, std::vector< double > contact_matrix)
Constructs a [ModelSEIRMixing](#) object.
- [ModelSEIRMixing](#) (const std::string &vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate, std::vector< double > contact_matrix)
Constructs a [ModelSEIRMixing](#) object.
- [ModelSEIRMixing](#)< TSeq > & [run](#) (epiworld_fast_uint ndays, int seed=-1)
Runs the simulation (after initialization)
- void [reset](#) ()
Reset the model.
- [Model](#)< TSeq > * [clone_ptr](#) ()
Advanced usage: Makes a copy of data and returns it as undeleted pointer.
- [ModelSEIRMixing](#)< TSeq > & [initial_states](#) (std::vector< double > proportions_, std::vector< int > queue_ = {})
Set the initial states of the model.
- void [set_contact_matrix](#) (std::vector< double > cmat)

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **EXPOSED** = 1
- static const int **INFECTED** = 2
- static const int **RECOVERED** = 3

Additional Inherited Members

16.40.1 Constructor & Destructor Documentation

16.40.1.1 ModelSEIRMixing() [1/2]

```
template<typename TSeq >
ModelSEIRMixing< TSeq >::ModelSEIRMixing (
    ModelSEIRMixing< TSeq > & model,
    const std::string & vname,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double avg_incubation_days,
    epiworld_double recovery_rate,
    std::vector< double > contact_matrix ) [inline]
```

Constructs a [ModelSEIRMixing](#) object.

Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.

Parameters

<i>model</i>	A reference to an existing ModelSEIRMixing object.
<i>vname</i>	The name of the ModelSEIRMixing object.
<i>n</i>	The number of entities in the model.
<i>prevalence</i>	The initial prevalence of the disease in the model.
<i>contact_rate</i>	The contact rate between entities in the model.
<i>transmission_rate</i>	The transmission rate of the disease in the model.
<i>avg_incubation_days</i>	The average incubation period of the disease in the model.
<i>recovery_rate</i>	The recovery rate of the disease in the model.
<i>contact_matrix</i>	The contact matrix between entities in the model. Specified in column-major order.
<i>model</i>	A <code>Model<TSeq></code> object where to set up the SIR.
<i>vname</i>	<code>std::string</code> Name of the virus
<i>prevalence</i>	Initial prevalence (proportion)
<i>contact_rate</i>	Average number of contacts (interactions) per step.
<i>transmission_rate</i>	Probability of transmission
<i>recovery_rate</i>	Probability of recovery

16.40.1.2 [ModelSEIRMixing\(\)](#) [2/2]

```
template<typename TSeq >
ModelSEIRMixing< TSeq >::ModelSEIRMixing (
    const std::string & vname,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double avg_incubation_days,
    epiworld_double recovery_rate,
    std::vector< double > contact_matrix ) [inline]
```

Constructs a [ModelSEIRMixing](#) object.

Parameters

<i>vname</i>	The name of the ModelSEIRMixing object.
<i>n</i>	The number of entities in the model.
<i>prevalence</i>	The initial prevalence of the disease in the model.
<i>contact_rate</i>	The contact rate between entities in the model.
<i>transmission_rate</i>	The transmission rate of the disease in the model.
<i>avg_incubation_days</i>	The average incubation period of the disease in the model.
<i>recovery_rate</i>	The recovery rate of the disease in the model.
<i>contact_matrix</i>	The contact matrix between entities in the model.

16.40.2 Member Function Documentation

16.40.2.1 clone_ptr()

```
template<typename TSeq >
Model< TSeq > * ModelSEIRMixing< TSeq >::clone_ptr [inline], [virtual]
```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

<i>copy</i>	
-------------	--

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.40.2.2 initial_states()

```
template<typename TSeq >
ModelSEIRMixing< TSeq > & ModelSEIRMixing< TSeq >::initial_states (
    std::vector< double > proportions_,
    std::vector< int > queue_ = {} ) [inline], [virtual]
```

Set the initial states of the model.

Parameters

<i>proportions_</i> ↔	Double vector with a single element:
—	<ul style="list-style-type: none"> The proportion of non-infected individuals who have recovered.

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.40.2.3 reset()

```
template<typename TSeq >
void ModelSEIRMixing< TSeq >::reset [inline], [virtual]
```

Reset the model.

Resetting the model will:

- clear the database
- restore the population (if `set_backup()` was called before)
- re-distribute tools
- re-distribute viruses
- set the date to 0

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

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

- `include/epiworld/models/seirmixing.hpp`

16.41 epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq > Class Template Reference

SEIR model with mixing, quarantine, and contact tracing.

```
#include <epiworld.hpp>
```

Inheritance diagram for epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >:



Collaboration diagram for epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >:



Public Member Functions

- [ModelSEIRMixingQuarantine](#) ([ModelSEIRMixingQuarantine](#)< TSeq > &model, const std::string &vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate, std::vector< double > contact_matrix, epiworld_double hospitalization_rate, epiworld_double hospitalization_period, epiworld_double days_undetected, epiworld_fast_int quarantine_period, epiworld_double quarantine_willingness, epiworld_double isolation_willingness, epiworld_fast_int isolation_period, epiworld_double contact_tracing_success_rate=1.0, epiworld_fast_uint contact_tracing_days_prior=4u)

Constructs a `ModelSEIRMixingQuarantine` object.

- `ModelSEIRMixingQuarantine` (const std::string &vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate, std::vector< double > contact_matrix, epiworld_double hospitalization_rate, epiworld_double hospitalization_period, epiworld_double days_undetected, epiworld_fast_int quarantine_↵_period, epiworld_double quarantine_willingness, epiworld_double isolation_willingness, epiworld_fast_int isolation_period, epiworld_double contact_tracing_success_rate=1.0, epiworld_fast_uint contact_tracing_↵_days_prior=4u)

Constructs a `ModelSEIRMixingQuarantine` object.

- `ModelSEIRMixingQuarantine< TSeq > & run` (epiworld_fast_uint ndays, int seed=-1)

Run the model simulation.

- void `reset` ()

Reset the model to initial state.

- `Model< TSeq > * clone_ptr` ()

Create a clone of this model.

- `ModelSEIRMixingQuarantine< TSeq > & initial_states` (std::vector< double > proportions_, std::vector< int > queue_={})

Set the initial states of the model.

- void `set_contact_matrix` (std::vector< double > cmat)

Set the contact matrix for population mixing.

- std::vector< double > `get_contact_matrix` () const

Get the current contact matrix.

- std::vector< size_t > `get_agent_quarantine_triggered` () const

Get the quarantine trigger status for all agents.

- std::vector< bool > `get_quarantine_willingness` () const

Get the quarantine willingness for all agents.

- std::vector< bool > `get_isolation_willingness` () const

Get the isolation willingness for all agents.

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **EXPOSED** = 1
- static const int **INFECTED** = 2
- static const int **ISOLATED** = 3
- static const int **DETECTED_HOSPITALIZED** = 4
- static const int **QUARANTINED_SUSCEPTIBLE** = 5
- static const int **QUARANTINED_EXPOSED** = 6
- static const int **ISOLATED_RECOVERED** = 7
- static const int **HOSPITALIZED** = 8
- static const int **RECOVERED** = 9
- static const size_t **QUARANTINE_PROCESS_INACTIVE** = 0u
- static const size_t **QUARANTINE_PROCESS_ACTIVE** = 1u
- static const size_t **QUARANTINE_PROCESS_DONE** = 2u

Additional Inherited Members

16.41.1 Detailed Description

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
class epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >
```

SEIR model with mixing, quarantine, and contact tracing.

This class implements a Susceptible-Exposed-Infected-Removed (SEIR) epidemiological model with additional features including:

- Population mixing based on contact matrices
- Quarantine measures for exposed contacts
- Isolation policies for detected infected individuals
- Contact tracing with configurable success rates
- Hospitalization of severe cases
- Individual willingness to comply with public health measures

The model supports 10 distinct states:

- Susceptible: Individuals who can become infected
- Exposed: Infected but not yet infectious (incubation period)
- Infected: Infectious individuals in the community
- Isolated: Detected infected individuals in self-isolation
- Detected Hospitalized: Hospitalized individuals who were contact-traced
- Quarantined Susceptible: Susceptible individuals in quarantine due to contact tracing
- Quarantined Exposed: Exposed individuals in quarantine due to contact tracing
- Isolated Recovered: Recovered individuals still in isolation
- Hospitalized: Individuals requiring hospital care
- Recovered: Individuals who have recovered and gained immunity

Template Parameters

<i>TSeq</i>	Type for genetic sequences (default: EPI_DEFAULT_TSEQ)
-------------	--

16.41.2 Constructor & Destructor Documentation

16.41.2.1 ModelSEIRMixingQuarantine() [1/2]

```
template<typename TSeq >
ModelSEIRMixingQuarantine< TSeq >::ModelSEIRMixingQuarantine (
    ModelSEIRMixingQuarantine< TSeq > & model,
    const std::string & vname,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double avg_incubation_days,
    epiworld_double recovery_rate,
    std::vector< double > contact_matrix,
    epiworld_double hospitalization_rate,
    epiworld_double hospitalization_period,
    epiworld_double days_undetected,
    epiworld_fast_int quarantine_period,
    epiworld_double quarantine_willingness,
    epiworld_double isolation_willingness,
    epiworld_fast_int isolation_period,
    epiworld_double contact_tracing_success_rate = 1.0,
    epiworld_fast_uint contact_tracing_days_prior = 4u ) [inline]
```

Constructs a [ModelSEIRMixingQuarantine](#) object.

Template for a Susceptible-Exposed-Infected-Removed (SEIR) model with mixing, quarantine, and contact tracing.

Parameters

<i>model</i>	A reference to an existing ModelSEIRMixingQuarantine object.
<i>vname</i>	The name of the ModelSEIRMixingQuarantine object.
<i>n</i>	The number of entities in the model.
<i>prevalence</i>	The initial prevalence of the disease in the model.
<i>contact_rate</i>	The contact rate between entities in the model.
<i>transmission_rate</i>	The transmission rate of the disease in the model.
<i>avg_incubation_days</i>	The average incubation period of the disease in the model.
<i>recovery_rate</i>	The recovery rate of the disease in the model.
<i>contact_matrix</i>	The contact matrix between entities in the model. Specified in column-major order.
<i>hospitalization_rate</i>	The rate at which infected individuals are hospitalized.
<i>hospitalization_period</i>	The average duration of hospitalization in days.
<i>days_undetected</i>	The average number of days an infected individual remains undetected.
<i>quarantine_period</i>	The duration of quarantine in days for exposed contacts.
<i>quarantine_willingness</i>	The proportion of individuals willing to comply with quarantine measures.
<i>isolation_willingness</i>	The proportion of individuals willing to self-isolate when detected.
<i>isolation_period</i>	The duration of isolation in days for detected infected individuals.
<i>contact_tracing_success_rate</i>	The probability of successfully identifying and tracing contacts (default: 1.0).
<i>contact_tracing_days_prior</i>	The number of days prior to detection for which contacts are traced (default: 4).
<i>model</i>	A ModelSEIRMixingQuarantine<TSeq> object where to set up the SEIR model.
<i>vname</i>	Name of the virus
<i>n</i>	Number of agents in the population
<i>prevalence</i>	Initial prevalence (proportion of infected individuals)

Parameters

<i>contact_rate</i>	Average number of contacts (interactions) per step
<i>transmission_rate</i>	Probability of transmission per contact
<i>avg_incubation_days</i>	Average incubation period in days
<i>recovery_rate</i>	Probability of recovery per day
<i>contact_matrix</i>	Contact matrix specifying mixing patterns between population groups
<i>hospitalization_rate</i>	Rate at which infected individuals are hospitalized
<i>hospitalization_period</i>	Average duration of hospitalization in days
<i>days_undetected</i>	Average number of days an infected individual remains undetected
<i>quarantine_period</i>	Duration of quarantine in days for exposed contacts
<i>quarantine_willingness</i>	Proportion of individuals willing to comply with quarantine
<i>isolation_willingness</i>	Proportion of individuals willing to self-isolate when detected
<i>isolation_period</i>	Duration of isolation in days for detected infected individuals
<i>contact_tracing_success_rate</i>	Probability of successfully identifying contacts during tracing
<i>contact_tracing_days_prior</i>	Number of days prior to detection for contact tracing

16.41.2.2 ModelSEIRMixingQuarantine() [2/2]

```
template<typename TSeq >
ModelSEIRMixingQuarantine< TSeq >::ModelSEIRMixingQuarantine (
    const std::string & vname,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double avg_incubation_days,
    epiworld_double recovery_rate,
    std::vector< double > contact_matrix,
    epiworld_double hospitalization_rate,
    epiworld_double hospitalization_period,
    epiworld_double days_undetected,
    epiworld_fast_int quarantine_period,
    epiworld_double quarantine_willingness,
    epiworld_double isolation_willingness,
    epiworld_fast_int isolation_period,
    epiworld_double contact_tracing_success_rate = 1.0,
    epiworld_fast_uint contact_tracing_days_prior = 4u ) [inline]
```

Constructs a [ModelSEIRMixingQuarantine](#) object.

Parameters

<i>vname</i>	The name of the ModelSEIRMixingQuarantine object.
<i>n</i>	The number of entities in the model.
<i>prevalence</i>	The initial prevalence of the disease in the model.
<i>contact_rate</i>	The contact rate between entities in the model.
<i>transmission_rate</i>	The transmission rate of the disease in the model.
<i>avg_incubation_days</i>	The average incubation period of the disease in the model.

Parameters

<i>recovery_rate</i>	The recovery rate of the disease in the model.
<i>contact_matrix</i>	The contact matrix between entities in the model.
<i>hospitalization_rate</i>	The rate at which infected individuals are hospitalized.
<i>hospitalization_period</i>	The average duration of hospitalization in days.
<i>days_undetected</i>	The average number of days an infected individual remains undetected.
<i>quarantine_period</i>	The duration of quarantine in days for exposed contacts.
<i>quarantine_willingness</i>	The proportion of individuals willing to comply with quarantine measures.
<i>isolation_willingness</i>	The proportion of individuals willing to self-isolate when detected.
<i>isolation_period</i>	The duration of isolation in days for detected infected individuals.
<i>contact_tracing_success_rate</i>	The probability of successfully identifying and tracing contacts (default: 1.0).
<i>contact_tracing_days_prior</i>	The number of days prior to detection for which contacts are traced (default: 4).

16.41.3 Member Function Documentation

16.41.3.1 `clone_ptr()`

```
template<typename TSeq >
Model< TSeq > * ModelSEIRMixingQuarantine< TSeq >::clone_ptr [inline], [virtual]
```

Create a clone of this model.

Returns

Pointer to a new model instance with the same configuration

Reimplemented from `epiworld::Model< EPI_DEFAULT_TSEQ >`.

16.41.3.2 `get_agent_quarantine_triggered()`

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
std::vector< size_t > epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >::get_agent_↵
quarantine_triggered ( ) const [inline]
```

Get the quarantine trigger status for all agents.

Returns

Vector indicating quarantine process status for each agent

16.41.3.3 get_contact_matrix()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
std::vector< double > epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >::get_contact_↵
matrix ( ) const [inline]
```

Get the current contact matrix.

Returns

Vector representing the contact matrix

16.41.3.4 get_isolation_willingness()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
std::vector< bool > epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >::get_isolation_↵
willingness ( ) const [inline]
```

Get the isolation willingness for all agents.

Returns

Vector of boolean values indicating each agent's willingness to self-isolate

16.41.3.5 get_quarantine_willingness()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
std::vector< bool > epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >::get_quarantine_↵
willingness ( ) const [inline]
```

Get the quarantine willingness for all agents.

Returns

Vector of boolean values indicating each agent's willingness to quarantine

16.41.3.6 initial_states()

```
template<typename TSeq >
ModelSEIRMixingQuarantine< TSeq > & ModelSEIRMixingQuarantine< TSeq >::initial_states (
    std::vector< double > proportions_,
    std::vector< int > queue_ = {} ) [inline], [virtual]
```

Set the initial states of the model.

Parameters

<i>proportions</i> ↔ —	Double vector with two elements: <ul style="list-style-type: none"> • [0]: The proportion of initially infected individuals who start in the exposed state. • [1]: The proportion of initially non-infected individuals who have recovered (immune).
<i>queue_</i>	Optional vector for queuing specifications (default: empty).

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.41.3.7 run()

```
template<typename TSeq >
ModelSEIRMixingQuarantine< TSeq > & ModelSEIRMixingQuarantine< TSeq >::run (
    epiworld_fast_uint ndays,
    int seed = -1 ) [inline], [virtual]
```

Run the model simulation.

Parameters

<i>ndays</i>	Number of days to simulate
<i>seed</i>	Random seed for reproducibility (default: -1 for random seed)

Returns

Reference to this model instance

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.41.3.8 set_contact_matrix()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
void epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >::set_contact_matrix (
    std::vector< double > cmat ) [inline]
```

Set the contact matrix for population mixing.

Parameters

<i>cmat</i>	Contact matrix specifying interaction rates between groups
-------------	--

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

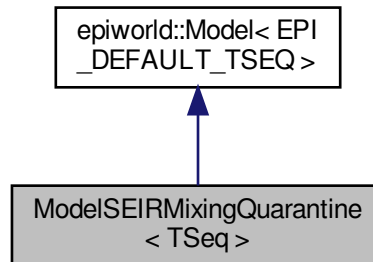
- epiworld.hpp

16.42 ModelSEIRMixingQuarantine< TSeq > Class Template Reference

SEIR model with mixing, quarantine, and contact tracing.

```
#include <seirmixingquarantine.hpp>
```

Inheritance diagram for ModelSEIRMixingQuarantine< TSeq >:



Collaboration diagram for ModelSEIRMixingQuarantine< TSeq >:



Public Member Functions

- [ModelSEIRMixingQuarantine](#) ([ModelSEIRMixingQuarantine](#)< TSeq > &model, const std::string &vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate, std::vector< double > contact_matrix, epiworld_double hospitalization_rate, epiworld_double hospitalization_period, epiworld_double days_undetected, epiworld_fast_int quarantine_period, epiworld_double quarantine_willingness, epiworld_double isolation_willingness, epiworld_fast_int isolation_period, epiworld_double contact_tracing_success_rate=1.0, epiworld_fast_uint contact_tracing_days_prior=4u)

Constructs a [ModelSEIRMixingQuarantine](#) object.

- [ModelSEIRMixingQuarantine](#) (const std::string &vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate, std::vector< double > contact_matrix, epiworld_double hospitalization_rate, epiworld_double hospitalization_period, epiworld_double days_undetected, epiworld_fast_int quarantine_↵_period, epiworld_double quarantine_willingness, epiworld_double isolation_willingness, epiworld_fast_int isolation_period, epiworld_double contact_tracing_success_rate=1.0, epiworld_fast_uint contact_tracing_↵_days_prior=4u)
Constructs a [ModelSEIRMixingQuarantine](#) object.
- [ModelSEIRMixingQuarantine< TSeq > & run](#) (epiworld_fast_uint ndays, int seed=-1)
Run the model simulation.
- void [reset](#) ()
Reset the model to initial state.
- [Model< TSeq > * clone_ptr](#) ()
Create a clone of this model.
- [ModelSEIRMixingQuarantine< TSeq > & initial_states](#) (std::vector< double > proportions_, std::vector< int > queue_={})
Set the initial states of the model.
- void [set_contact_matrix](#) (std::vector< double > cmat)
Set the contact matrix for population mixing.
- std::vector< double > [get_contact_matrix](#) () const
Get the current contact matrix.
- std::vector< size_t > [get_agent_quarantine_triggered](#) () const
Get the quarantine trigger status for all agents.
- std::vector< bool > [get_quarantine_willingness](#) () const
Get the quarantine willingness for all agents.
- std::vector< bool > [get_isolation_willingness](#) () const
Get the isolation willingness for all agents.

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **EXPOSED** = 1
- static const int **INFECTED** = 2
- static const int **ISOLATED** = 3
- static const int **DETECTED_HOSPITALIZED** = 4
- static const int **QUARANTINED_SUSCEPTIBLE** = 5
- static const int **QUARANTINED_EXPOSED** = 6
- static const int **ISOLATED_RECOVERED** = 7
- static const int **HOSPITALIZED** = 8
- static const int **RECOVERED** = 9
- static const size_t **QUARANTINE_PROCESS_INACTIVE** = 0u
- static const size_t **QUARANTINE_PROCESS_ACTIVE** = 1u
- static const size_t **QUARANTINE_PROCESS_DONE** = 2u

Additional Inherited Members

16.42.1 Detailed Description

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
class ModelSEIRMixingQuarantine< TSeq >
```

SEIR model with mixing, quarantine, and contact tracing.

This class implements a Susceptible-Exposed-Infected-Removed (SEIR) epidemiological model with additional features including:

- Population mixing based on contact matrices
- Quarantine measures for exposed contacts
- Isolation policies for detected infected individuals
- Contact tracing with configurable success rates
- Hospitalization of severe cases
- Individual willingness to comply with public health measures

The model supports 10 distinct states:

- Susceptible: Individuals who can become infected
- Exposed: Infected but not yet infectious (incubation period)
- Infected: Infectious individuals in the community
- Isolated: Detected infected individuals in self-isolation
- Detected Hospitalized: Hospitalized individuals who were contact-traced
- Quarantined Susceptible: Susceptible individuals in quarantine due to contact tracing
- Quarantined Exposed: Exposed individuals in quarantine due to contact tracing
- Isolated Recovered: Recovered individuals still in isolation
- Hospitalized: Individuals requiring hospital care
- Recovered: Individuals who have recovered and gained immunity

Template Parameters

<i>TSeq</i>	Type for genetic sequences (default: EPI_DEFAULT_TSEQ)
-------------	--

16.42.2 Constructor & Destructor Documentation

16.42.2.1 ModelSEIRMixingQuarantine() [1/2]

```

template<typename TSeq >
ModelSEIRMixingQuarantine< TSeq >::ModelSEIRMixingQuarantine (
    ModelSEIRMixingQuarantine< TSeq > & model,
    const std::string & vname,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double avg_incubation_days,
    epiworld_double recovery_rate,
    std::vector< double > contact_matrix,
    epiworld_double hospitalization_rate,
    epiworld_double hospitalization_period,
    epiworld_double days_undetected,
    epiworld_fast_int quarantine_period,
    epiworld_double quarantine_willingness,
    epiworld_double isolation_willingness,
    epiworld_fast_int isolation_period,
    epiworld_double contact_tracing_success_rate = 1.0,
    epiworld_fast_uint contact_tracing_days_prior = 4u ) [inline]

```

Constructs a [ModelSEIRMixingQuarantine](#) object.

Template for a Susceptible-Exposed-Infected-Removed (SEIR) model with mixing, quarantine, and contact tracing.

Parameters

<i>model</i>	A reference to an existing ModelSEIRMixingQuarantine object.
<i>vname</i>	The name of the ModelSEIRMixingQuarantine object.
<i>n</i>	The number of entities in the model.
<i>prevalence</i>	The initial prevalence of the disease in the model.
<i>contact_rate</i>	The contact rate between entities in the model.
<i>transmission_rate</i>	The transmission rate of the disease in the model.
<i>avg_incubation_days</i>	The average incubation period of the disease in the model.
<i>recovery_rate</i>	The recovery rate of the disease in the model.
<i>contact_matrix</i>	The contact matrix between entities in the model. Specified in column-major order.
<i>hospitalization_rate</i>	The rate at which infected individuals are hospitalized.
<i>hospitalization_period</i>	The average duration of hospitalization in days.
<i>days_undetected</i>	The average number of days an infected individual remains undetected.
<i>quarantine_period</i>	The duration of quarantine in days for exposed contacts.
<i>quarantine_willingness</i>	The proportion of individuals willing to comply with quarantine measures.
<i>isolation_willingness</i>	The proportion of individuals willing to self-isolate when detected.
<i>isolation_period</i>	The duration of isolation in days for detected infected individuals.
<i>contact_tracing_success_rate</i>	The probability of successfully identifying and tracing contacts (default: 1.0).
<i>contact_tracing_days_prior</i>	The number of days prior to detection for which contacts are traced (default: 4).
<i>model</i>	A ModelSEIRMixingQuarantine<TSeq> object where to set up the SEIR model.
<i>vname</i>	Name of the virus
<i>n</i>	Number of agents in the population
<i>prevalence</i>	Initial prevalence (proportion of infected individuals)

Parameters

<i>contact_rate</i>	Average number of contacts (interactions) per step
<i>transmission_rate</i>	Probability of transmission per contact
<i>avg_incubation_days</i>	Average incubation period in days
<i>recovery_rate</i>	Probability of recovery per day
<i>contact_matrix</i>	Contact matrix specifying mixing patterns between population groups
<i>hospitalization_rate</i>	Rate at which infected individuals are hospitalized
<i>hospitalization_period</i>	Average duration of hospitalization in days
<i>days_undetected</i>	Average number of days an infected individual remains undetected
<i>quarantine_period</i>	Duration of quarantine in days for exposed contacts
<i>quarantine_willingness</i>	Proportion of individuals willing to comply with quarantine
<i>isolation_willingness</i>	Proportion of individuals willing to self-isolate when detected
<i>isolation_period</i>	Duration of isolation in days for detected infected individuals
<i>contact_tracing_success_rate</i>	Probability of successfully identifying contacts during tracing
<i>contact_tracing_days_prior</i>	Number of days prior to detection for contact tracing

16.42.2.2 `ModelSEIRMixingQuarantine()` [2/2]

```
template<typename TSeq >
ModelSEIRMixingQuarantine< TSeq >::ModelSEIRMixingQuarantine (
    const std::string & vname,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double avg_incubation_days,
    epiworld_double recovery_rate,
    std::vector< double > contact_matrix,
    epiworld_double hospitalization_rate,
    epiworld_double hospitalization_period,
    epiworld_double days_undetected,
    epiworld_fast_int quarantine_period,
    epiworld_double quarantine_willingness,
    epiworld_double isolation_willingness,
    epiworld_fast_int isolation_period,
    epiworld_double contact_tracing_success_rate = 1.0,
    epiworld_fast_uint contact_tracing_days_prior = 4u ) [inline]
```

Constructs a `ModelSEIRMixingQuarantine` object.

Parameters

<i>vname</i>	The name of the <code>ModelSEIRMixingQuarantine</code> object.
<i>n</i>	The number of entities in the model.
<i>prevalence</i>	The initial prevalence of the disease in the model.
<i>contact_rate</i>	The contact rate between entities in the model.
<i>transmission_rate</i>	The transmission rate of the disease in the model.
<i>avg_incubation_days</i>	The average incubation period of the disease in the model.

Parameters

<i>recovery_rate</i>	The recovery rate of the disease in the model.
<i>contact_matrix</i>	The contact matrix between entities in the model.
<i>hospitalization_rate</i>	The rate at which infected individuals are hospitalized.
<i>hospitalization_period</i>	The average duration of hospitalization in days.
<i>days_undetected</i>	The average number of days an infected individual remains undetected.
<i>quarantine_period</i>	The duration of quarantine in days for exposed contacts.
<i>quarantine_willingness</i>	The proportion of individuals willing to comply with quarantine measures.
<i>isolation_willingness</i>	The proportion of individuals willing to self-isolate when detected.
<i>isolation_period</i>	The duration of isolation in days for detected infected individuals.
<i>contact_tracing_success_rate</i>	The probability of successfully identifying and tracing contacts (default: 1.0).
<i>contact_tracing_days_prior</i>	The number of days prior to detection for which contacts are traced (default: 4).

16.42.3 Member Function Documentation

16.42.3.1 clone_ptr()

```
template<typename TSeq >
Model< TSeq > * ModelSEIRMixingQuarantine< TSeq >::clone_ptr [inline], [virtual]
```

Create a clone of this model.

Returns

Pointer to a new model instance with the same configuration

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.42.3.2 get_agent_quarantine_triggered()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
std::vector< size_t > ModelSEIRMixingQuarantine< TSeq >::get_agent_quarantine_triggered ( )
const [inline]
```

Get the quarantine trigger status for all agents.

Returns

Vector indicating quarantine process status for each agent

16.42.3.3 `get_contact_matrix()`

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
std::vector< double > ModelSEIRMixingQuarantine< TSeq >::get_contact_matrix ( ) const [inline]
```

Get the current contact matrix.

Returns

Vector representing the contact matrix

16.42.3.4 `get_isolation_willingness()`

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
std::vector< bool > ModelSEIRMixingQuarantine< TSeq >::get_isolation_willingness ( ) const
[inline]
```

Get the isolation willingness for all agents.

Returns

Vector of boolean values indicating each agent's willingness to self-isolate

16.42.3.5 `get_quarantine_willingness()`

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
std::vector< bool > ModelSEIRMixingQuarantine< TSeq >::get_quarantine_willingness ( ) const
[inline]
```

Get the quarantine willingness for all agents.

Returns

Vector of boolean values indicating each agent's willingness to quarantine

16.42.3.6 `initial_states()`

```
template<typename TSeq >
ModelSEIRMixingQuarantine< TSeq > & ModelSEIRMixingQuarantine< TSeq >::initial_states (
    std::vector< double > proportions_,
    std::vector< int > queue_ = {} ) [inline], [virtual]
```

Set the initial states of the model.

Parameters

<i>proportions</i> ↔ —	Double vector with two elements: <ul style="list-style-type: none"> • [0]: The proportion of initially infected individuals who start in the exposed state. • [1]: The proportion of initially non-infected individuals who have recovered (immune).
<i>queue_</i>	Optional vector for queuing specifications (default: empty).

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.42.3.7 run()

```
template<typename TSeq >
ModelSEIRMixingQuarantine< TSeq > & ModelSEIRMixingQuarantine< TSeq >::run (
    epiworld_fast_uint ndays,
    int seed = -1 ) [inline], [virtual]
```

Run the model simulation.

Parameters

<i>ndays</i>	Number of days to simulate
<i>seed</i>	Random seed for reproducibility (default: -1 for random seed)

Returns

Reference to this model instance

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.42.3.8 set_contact_matrix()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
void ModelSEIRMixingQuarantine< TSeq >::set_contact_matrix (
    std::vector< double > cmat ) [inline]
```

Set the contact matrix for population mixing.

Parameters

<i>cmat</i>	Contact matrix specifying interaction rates between groups
-------------	--

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

- include/epiworld/models/[seirmixingquarantine.hpp](#)

16.43 epiworld::epimodels::ModelSIR< TSeq > Class Template Reference

Template for a Susceptible-Infected-Removed (SIR) model.

```
#include <epiworld.hpp>
```

Inheritance diagram for epiworld::epimodels::ModelSIR< TSeq >:



Collaboration diagram for epiworld::epimodels::ModelSIR< TSeq >:



Public Member Functions

- **ModelSIR** ([ModelSIR](#)< TSeq > &model, const std::string &vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double recovery_rate)
- **ModelSIR** (const std::string &vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double recovery_rate)
- [ModelSIR](#)< TSeq > & [initial_states](#) (std::vector< double > proportions_, std::vector< int > queue_={})

Set the initial states of the model.

Additional Inherited Members

16.43.1 Detailed Description

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
class epiworld::epimodels::ModelSIR< TSeq >
```

Template for a Susceptible-Infected-Removed (SIR) model.

Parameters

<i>model</i>	A Model<TSeq> object where to set up the SIR.
<i>vname</i>	std::string Name of the virus
<i>initial_prevalence</i>	epiworld_double Initial prevalence
<i>initial_efficiency</i>	epiworld_double Initial susceptibility_reduction of the immune system
<i>initial_recovery</i>	epiworld_double Initial recovery_rate rate of the immune system

16.43.2 Member Function Documentation

16.43.2.1 initial_states()

```
template<typename TSeq >
ModelSIR< TSeq > & ModelSIR< TSeq >::initial_states (
    std::vector< double > proportions_,
    std::vector< int > queue_ = {} ) [inline], [virtual]
```

Set the initial states of the model.

Parameters

<i>proportions_</i>	Double vector with a single element:
—	<ul style="list-style-type: none"> The proportion of non-infected individuals who have recovered.

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

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

- epiworld.hpp

16.44 ModelSIR< TSeq > Class Template Reference

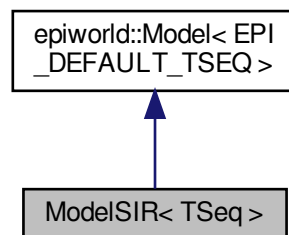
Template for a Susceptible-Infected-Removed (SIR) model.

```
#include <sir.hpp>
```

Inheritance diagram for ModelSIR< TSeq >:



Collaboration diagram for ModelSIR< TSeq >:



Public Member Functions

- **ModelSIR** ([ModelSIR](#)< TSeq > &model, const std::string &vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double recovery_rate)
- **ModelSIR** (const std::string &vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double recovery_rate)
- [ModelSIR](#)< TSeq > & [initial_states](#) (std::vector< double > proportions_, std::vector< int > queue_={})

Set the initial states of the model.

Additional Inherited Members

16.44.1 Detailed Description

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
class ModelSIR< TSeq >
```

Template for a Susceptible-Infected-Removed (SIR) model.

Parameters

<i>model</i>	A Model<TSeq> object where to set up the SIR.
<i>vname</i>	std::string Name of the virus
<i>initial_prevalence</i>	epiworld_double Initial prevalence
<i>initial_efficacy</i>	epiworld_double Initial susceptibility_reduction of the immune system
<i>initial_recovery</i>	epiworld_double Initial recovery_rate rate of the immune system

16.44.2 Member Function Documentation

16.44.2.1 initial_states()

```
template<typename TSeq >
ModelSIR< TSeq > & ModelSIR< TSeq >::initial_states (
    std::vector< double > proportions_,
    std::vector< int > queue_ = {} ) [inline], [virtual]
```

Set the initial states of the model.

Parameters

<i>proportions_</i> ↔	Double vector with a single element:
—	<ul style="list-style-type: none"> The proportion of non-infected individuals who have recovered.

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

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

- include/epiworld/models/sir.hpp

16.45 epiworld::epimodels::ModelSIRCONN< TSeq > Class Template Reference

Inheritance diagram for epiworld::epimodels::ModelSIRCONN< TSeq >:



Collaboration diagram for epiworld::epimodels::ModelSIRCONN< TSeq >:



Public Member Functions

- [ModelSIRCONN](#) ([ModelSIRCONN](#)< TSeq > &model, const std::string &vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double recovery_rate)
Template for a Susceptible-Infected-Removed (SIR) model.
- **ModelSIRCONN** (const std::string &vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double recovery_rate)
- [ModelSIRCONN](#)< TSeq > & [run](#) (epiworld_fast_uint ndays, int seed=-1)
Runs the simulation (after initialization)
- void [reset](#) ()
Reset the model.
- [Model](#)< TSeq > * [clone_ptr](#) ()

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

- `ModelSIRCONN< TSeq > & initial_states` (`std::vector< double > proportions_`, `std::vector< int > queue_` → `_={}`)
Set the initial states of the model.
- `size_t get_n_infected` () const
Get the infected individuals.
- `std::vector< double > generation_time_expected` (int `max_days=200`, int `max_contacts=200`) const

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **INFECTED** = 1
- static const int **RECOVERED** = 2

Additional Inherited Members

16.45.1 Constructor & Destructor Documentation

16.45.1.1 ModelSIRCONN()

```
template<typename TSeq >
ModelSIRCONN< TSeq >::ModelSIRCONN (
    ModelSIRCONN< TSeq > & model,
    const std::string & vname,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double recovery_rate ) [inline]
```

Template for a Susceptible-Infected-Removed (SIR) model.

Parameters

<i>model</i>	A Model<TSeq> object where to set up the SIR.
<i>vname</i>	std::string Name of the virus
<i>prevalence</i>	Initial prevalence (proportion)
<i>contact_rate</i>	Average number of contacts (interactions) per step.
<i>transmission_rate</i>	Probability of transmission
<i>recovery_rate</i>	Probability of recovery

16.45.2 Member Function Documentation

16.45.2.1 clone_ptr()

```
template<typename TSeq >
Model< TSeq > * ModelSIRCONN< TSeq >::clone_ptr [inline], [virtual]
```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

<i>copy</i>	
-------------	--

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.45.2.2 get_n_infected()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
size_t epiworld::epimodels::ModelSIRCONN< TSeq >::get_n_infected ( ) const [inline]
```

Get the infected individuals.

Returns

std::vector< epiworld::Agent<TSeq> * >

16.45.2.3 initial_states()

```
template<typename TSeq >
ModelSIRCONN< TSeq > & ModelSIRCONN< TSeq >::initial_states (
    std::vector< double > proportions_,
    std::vector< int > queue_ = {} ) [inline], [virtual]
```

Set the initial states of the model.

Parameters

<i>proportions_</i> ↔	Double vector with a single element:
—	<ul style="list-style-type: none"> The proportion of non-infected individuals who have recovered.

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.45.2.4 reset()

```
template<typename TSeq >
void ModelSIRCONN< TSeq >::reset [inline], [virtual]
```


Reset the model.

Resetting the model will:

- clear the database
- restore the population (if `set_backup()` was called before)
- re-distribute tools
- re-distribute viruses
- set the date to 0

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

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

- `epiworld.hpp`

16.46 ModelSIRCONN< TSeq > Class Template Reference

Inheritance diagram for ModelSIRCONN< TSeq >:



Collaboration diagram for ModelSIRCONN< TSeq >:



Public Member Functions

- [ModelSIRCONN](#) ([ModelSIRCONN](#)< TSeq > &model, const std::string &vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double recovery_rate)
Template for a Susceptible-Infected-Removed (SIR) model.
- **ModelSIRCONN** (const std::string &vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double recovery_rate)
- [ModelSIRCONN](#)< TSeq > & [run](#) (epiworld_fast_uint ndays, int seed=-1)
Runs the simulation (after initialization)
- void [reset](#) ()
Reset the model.
- [Model](#)< TSeq > * [clone_ptr](#) ()
Advanced usage: Makes a copy of data and returns it as undeleted pointer.
- [ModelSIRCONN](#)< TSeq > & [initial_states](#) (std::vector< double > proportions_, std::vector< int > queue_ = {})
Set the initial states of the model.
- size_t [get_n_infected](#) () const
Get the infected individuals.
- std::vector< double > **generation_time_expected** (int max_days=200, int max_contacts=200) const

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **INFECTED** = 1
- static const int **RECOVERED** = 2

Additional Inherited Members

16.46.1 Constructor & Destructor Documentation

16.46.1.1 ModelSIRCONN()

```
template<typename TSeq >
ModelSIRCONN< TSeq >::ModelSIRCONN (
    ModelSIRCONN< TSeq > & model,
    const std::string & vname,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double recovery_rate ) [inline]
```

Template for a Susceptible-Infected-Removed (SIR) model.

Parameters

<i>model</i>	A Model<TSeq> object where to set up the SIR.
<i>vname</i>	std::string Name of the virus
<i>prevalence</i>	Initial prevalence (proportion)
<i>contact_rate</i>	Average number of contacts (interactions) per step.
<i>transmission_rate</i>	Probability of transmission
<i>recovery_rate</i>	Probability of recovery

16.46.2 Member Function Documentation

16.46.2.1 clone_ptr()

```
template<typename TSeq >
Model< TSeq > * ModelSIRCONN< TSeq >::clone_ptr [inline], [virtual]
```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

<i>copy</i>	
-------------	--

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.46.2.2 get_n_infected()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
size_t ModelSIRCONN< TSeq >::get_n_infected ( ) const [inline]
```

Get the infected individuals.

Returns

`std::vector< epiworld::Agent<TSeq> * >`

16.46.2.3 initial_states()

```
template<typename TSeq >
ModelSIRCONN< TSeq > & ModelSIRCONN< TSeq >::initial_states (
    std::vector< double > proportions_,
    std::vector< int > queue_ = {} ) [inline], [virtual]
```

Set the initial states of the model.

Parameters

<i>proportions_</i>	Double vector with a single element:
—	<ul style="list-style-type: none"> The proportion of non-infected individuals who have recovered.

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.46.2.4 reset()

```
template<typename TSeq >
void ModelSIRCONN< TSeq >::reset [inline], [virtual]
```

Reset the model.

Resetting the model will:

- clear the database
- restore the population (if `set_backup()` was called before)
- re-distribute tools
- re-distribute viruses
- set the date to 0

Reimplemented from `epiworld::Model< EPI_DEFAULT_TSEQ >`.

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

- `include/epiworld/models/sirconnected.hpp`

16.47 epiworld::epimodels::ModelSIRD< TSeq > Class Template Reference

Template for a Susceptible-Infected-Removed-Deceased (SIRD) model.

```
#include <epiworld.hpp>
```

Inheritance diagram for `epiworld::epimodels::ModelSIRD< TSeq >`:



Collaboration diagram for epiworld::epimodels::ModelSIRD< TSeq >:



Public Member Functions

- [ModelSIRD< TSeq >](#) & [initial_states](#) (std::vector< double > proportions_, std::vector< int > queue_={})
Set the initial states of the model.
- [ModelSIRD](#) ([ModelSIRD< TSeq >](#) &model, const std::string &vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double recovery_rate, epiworld_double death_rate)
Constructs a new SIRD model with the given parameters.
- **ModelSIRD** (const std::string &vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double recovery_rate, epiworld_double death_rate)

Additional Inherited Members

16.47.1 Detailed Description

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
class epiworld::epimodels::ModelSIRD< TSeq >
```

Template for a Susceptible-Infected-Removed-Deceased (SIRD) model.

16.47.2 Constructor & Destructor Documentation

16.47.2.1 ModelSIRD()

```
template<typename TSeq >
ModelSIRD< TSeq >::ModelSIRD (
    ModelSIRD< TSeq > & model,
    const std::string & vname,
    epiworld_double prevalence,
    epiworld_double transmission_rate,
    epiworld_double recovery_rate,
    epiworld_double death_rate ) [inline]
```

Constructs a new SIRD model with the given parameters.

Parameters

<i>model</i>	The SIRD model to copy from.
<i>vname</i>	The name of the vertex associated with this model.
<i>prevalence</i>	The initial prevalence of the disease in the population.
<i>transmission_rate</i>	The rate at which the disease spreads from infected to susceptible individuals.
<i>recovery_rate</i>	The rate at which infected individuals recover and become immune.
<i>death_rate</i>	The rate at which infected individuals die.

16.47.3 Member Function Documentation

16.47.3.1 initial_states()

```
template<typename TSeq >
ModelSIRD< TSeq > & ModelSIRD< TSeq >::initial_states (
    std::vector< double > proportions_,
    std::vector< int > queue_ = {} ) [inline], [virtual]
```

Set the initial states of the model.

Parameters

<i>proportions_</i>	Double vector with two elements:
—	<ul style="list-style-type: none"> • The proportion of non-infected individuals who have recovered. • The proportion of non-infected individuals who have died.

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

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

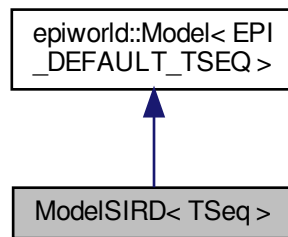
- [epiworld.hpp](#)

16.48 ModelSIRD< TSeq > Class Template Reference

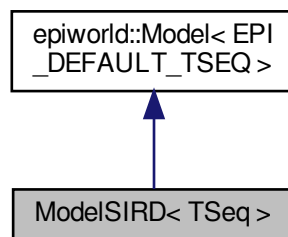
Template for a Susceptible-Infected-Removed-Deceased (SIRD) model.

```
#include <sird.hpp>
```

Inheritance diagram for ModelSIRD< TSeq >:



Collaboration diagram for ModelSIRD< TSeq >:



Public Member Functions

- [ModelSIRD](#) < TSeq > & [initial_states](#) (std::vector< double > proportions_, std::vector< int > queue_={})
Set the initial states of the model.
- [ModelSIRD](#) ([ModelSIRD](#) < TSeq > &model, const std::string &vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double recovery_rate, epiworld_double death_rate)
Constructs a new SIRD model with the given parameters.
- **ModelSIRD** (const std::string &vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double recovery_rate, epiworld_double death_rate)

Additional Inherited Members

16.48.1 Detailed Description

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
class ModelSIRD< TSeq >
```

Template for a Susceptible-Infected-Removed-Deceased (SIRD) model.

16.48.2 Constructor & Destructor Documentation

16.48.2.1 ModelSIRD()

```
template<typename TSeq >
ModelSIRD< TSeq >::ModelSIRD (
    ModelSIRD< TSeq > & model,
    const std::string & vname,
    epiworld_double prevalence,
    epiworld_double transmission_rate,
    epiworld_double recovery_rate,
    epiworld_double death_rate ) [inline]
```

Constructs a new SIRD model with the given parameters.

Parameters

<i>model</i>	The SIRD model to copy from.
<i>vname</i>	The name of the vertex associated with this model.
<i>prevalence</i>	The initial prevalence of the disease in the population.
<i>transmission_rate</i>	The rate at which the disease spreads from infected to susceptible individuals.
<i>recovery_rate</i>	The rate at which infected individuals recover and become immune.
<i>death_rate</i>	The rate at which infected individuals die.

16.48.3 Member Function Documentation

16.48.3.1 initial_states()

```
template<typename TSeq >
ModelSIRD< TSeq > & ModelSIRD< TSeq >::initial_states (
    std::vector< double > proportions_,
    std::vector< int > queue_ = {} ) [inline], [virtual]
```

Set the initial states of the model.

Parameters

<i>proportions_</i>	Double vector with two elements:
—	<ul style="list-style-type: none"> • The proportion of non-infected individuals who have recovered. • The proportion of non-infected individuals who have died.

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

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

- include/epiworld/models/sird.hpp

16.49 epiworld::epimodels::ModelSIRDCONN< TSeq > Class Template Reference

Inheritance diagram for epiworld::epimodels::ModelSIRDCONN< TSeq >:



Collaboration diagram for epiworld::epimodels::ModelSIRDCONN< TSeq >:



Public Member Functions

- [ModelSIRDCONN](#) ([ModelSIRDCONN](#)< TSeq > &model, const std::string &vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double recovery_rate, epiworld_double death_rate)

Template for a Susceptible-Infected-Removed (SIR) model.

- **ModelSIRDCONN** (const std::string &vname, ewiworld_fast_uint n, ewiworld_double prevalence, ewiworld_double contact_rate, ewiworld_double transmission_rate, ewiworld_double recovery_rate, ewiworld_double death_rate)
- **ModelSIRDCONN**< TSeq > & **run** (ewiworld_fast_uint ndays, int seed=-1)
Runs the simulation (after initialization)
- void **reset** ()
Reset the model.
- **Model**< TSeq > * **clone_ptr** ()
Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **INFECTED** = 1
- static const int **RECOVERED** = 2
- static const int **DECEASED** = 3

Additional Inherited Members

16.49.1 Constructor & Destructor Documentation

16.49.1.1 ModelSIRDCONN()

```
template<typename TSeq >
ModelSIRDCONN< TSeq >::ModelSIRDCONN (
    ModelSIRDCONN< TSeq > & model,
    const std::string & vname,
    ewiworld_fast_uint n,
    ewiworld_double prevalence,
    ewiworld_double contact_rate,
    ewiworld_double transmission_rate,
    ewiworld_double recovery_rate,
    ewiworld_double death_rate ) [inline]
```

Template for a Susceptible-Infected-Removed (SIR) model.

Parameters

<i>model</i>	A Model<TSeq> object where to set up the SIR.
<i>vname</i>	std::string Name of the virus
<i>prevalence</i>	Initial prevalence (proportion)
<i>contact_rate</i>	Average number of contacts (interactions) per step.
<i>transmission_rate</i>	Probability of transmission
<i>recovery_rate</i>	Probability of recovery
<i>death_rate</i>	Probability of death

16.49.2 Member Function Documentation

16.49.2.1 clone_ptr()

```
template<typename TSeq >
Model< TSeq > * ModelSIRDCONN< TSeq >::clone_ptr [inline], [virtual]
```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

<i>copy</i>	
-------------	--

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.49.2.2 reset()

```
template<typename TSeq >
void ModelSIRDCONN< TSeq >::reset [inline], [virtual]
```

Reset the model.

Resetting the model will:

- clear the database
- restore the population (if `set_backup()` was called before)
- re-distribute tools
- re-distribute viruses
- set the date to 0

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

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

- `epiworld.hpp`

16.50 ModelSIRDConn< TSeq > Class Template Reference

Inheritance diagram for ModelSIRDConn< TSeq >:



Collaboration diagram for ModelSIRDConn< TSeq >:



Public Member Functions

- [ModelSIRDConn](#) ([ModelSIRDConn](#)< TSeq > &model, const std::string &vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double recovery_rate, epiworld_double death_rate)
Template for a Susceptible-Infected-Removed (SIR) model.
- **ModelSIRDConn** (const std::string &vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double recovery_rate, epiworld_double death_rate)
- [ModelSIRDConn](#)< TSeq > & [run](#) (epiworld_fast_uint ndays, int seed=-1)
Runs the simulation (after initialization)
- void [reset](#) ()
Reset the model.
- [Model](#)< TSeq > * [clone_ptr](#) ()
Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **INFECTED** = 1
- static const int **RECOVERED** = 2
- static const int **DECEASED** = 3

Additional Inherited Members

16.50.1 Constructor & Destructor Documentation

16.50.1.1 ModelSIRDCONN()

```
template<typename TSeq >
ModelSIRDCONN< TSeq >::ModelSIRDCONN (
    ModelSIRDCONN< TSeq > & model,
    const std::string & vname,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double recovery_rate,
    epiworld_double death_rate ) [inline]
```

Template for a Susceptible-Infected-Removed (SIR) model.

Parameters

<i>model</i>	A Model<TSeq> object where to set up the SIR.
<i>vname</i>	std::string Name of the virus
<i>prevalence</i>	Initial prevalence (proportion)
<i>contact_rate</i>	Average number of contacts (interactions) per step.
<i>transmission_rate</i>	Probability of transmission
<i>recovery_rate</i>	Probability of recovery
<i>death_rate</i>	Probability of death

16.50.2 Member Function Documentation

16.50.2.1 clone_ptr()

```
template<typename TSeq >
Model< TSeq > * ModelSIRDCONN< TSeq >::clone_ptr [inline], [virtual]
```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

[copy](#)

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.50.2.2 reset()

```
template<typename TSeq >
void ModelSIRDCONN< TSeq >::reset [inline], [virtual]
```

Reset the model.

Resetting the model will:

- clear the database
- restore the population (if `set_backup()` was called before)
- re-distribute tools
- re-distribute viruses
- set the date to 0

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

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

- `include/epiworld/models/sirdconnected.hpp`

16.51 epiworld::epimodels::ModelSIRLogit< TSeq > Class Template Reference

Template for a Susceptible-Infected-Removed (SIR) model.

```
#include <epiworld.hpp>
```

Inheritance diagram for `epiworld::epimodels::ModelSIRLogit< TSeq >`:



Collaboration diagram for `epiworld::epimodels::ModelSIRLogit< TSeq >`:



Public Member Functions

- [ModelSIRLogit](#) ([ModelSIRLogit](#)< TSeq > &model, const std::string &vname, double *data, size_t ncols, std::vector< double > coefs_infect, std::vector< double > coefs_recover, std::vector< size_t > coef_infect_cols, std::vector< size_t > coef_recover_cols, epiworld_double transmission_rate, epiworld_double recovery_rate, epiworld_double prevalence)

Template for a Susceptible-Infected-Removed (SIR) model.

- **ModelSIRLogit** (const std::string &vname, double *data, size_t ncols, std::vector< double > coefs_infect, std::vector< double > coefs_recover, std::vector< size_t > coef_infect_cols, std::vector< size_t > coef_recover_cols, epiworld_double transmission_rate, epiworld_double recovery_rate, epiworld_double prevalence)
- [ModelSIRLogit](#)< TSeq > & [run](#) (epiworld_fast_uint ndays, int seed=-1)
Runs the simulation (after initialization)
- [Model](#)< TSeq > * [clone_ptr](#) ()
Advanced usage: Makes a copy of data and returns it as undeleted pointer.
- void [reset](#) ()
Reset the model.

Public Attributes

- std::vector< double > **coefs_infect**
- std::vector< double > **coefs_recover**
- std::vector< size_t > **coef_infect_cols**
- std::vector< size_t > **coef_recover_cols**

Additional Inherited Members

16.51.1 Detailed Description

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
class epiworld::epimodels::ModelSIRLogit< TSeq >
```

Template for a Susceptible-Infected-Removed (SIR) model.

In this model, infection and recovery probabilities are computed using a logit model. Particularly, the probability of infection is computed as:

$$\frac{1}{1 + \exp(-(\beta_0 E_i + \sum_{i=1}^n \beta_i x_i))}$$

where β_0 is the exposure coefficient and E_i is the exposure number, β_i are the coefficients for the features x_i of the agents, and n is the number of features. The probability of recovery is computed as:

$$\frac{1}{1 + \exp(-(\sum_{i=1}^n \beta_i x_i))}$$

where β_i are the coefficients for the features x_i of the agents, and n is the number of features.

Parameters

<i>TSeq</i>	Type of the sequence (e.g. <code>std::vector</code> , <code>std::deque</code>)
-------------	---

16.51.2 Constructor & Destructor Documentation

16.51.2.1 ModelSIRLogit()

```
template<typename TSeq >
ModelSIRLogit< TSeq >::ModelSIRLogit (
    ModelSIRLogit< TSeq > & model,
    const std::string & vname,
    double * data,
    size_t ncols,
    std::vector< double > coefs_infect,
    std::vector< double > coefs_recover,
    std::vector< size_t > coef_infect_cols,
    std::vector< size_t > coef_recover_cols,
    epiworld_double transmission_rate,
    epiworld_double recovery_rate,
    epiworld_double prevalence ) [inline]
```

Template for a Susceptible-Infected-Removed (SIR) model.

Parameters

<i>vname</i>	Name of the virus.
<i>coefs_infect</i>	Double ptr. Infection coefficients.
<i>coefs_recover</i>	Double ptr. Recovery coefficients.
<i>ncoen_infect</i>	Unsigned int. Number of infection coefficients.
<i>ncoen_recover</i>	Unsigned int. Number of recovery coefficients.
<i>coef_infect_cols</i>	Vector<unsigned int>. Ids of infection vars.
<i>coef_recover_cols</i>	Vector<unsigned int>. Ids of recover vars.

Parameters

<i>model</i>	A Model<TSeq> object where to set up the SIR.
<i>vname</i>	std::string Name of the virus
<i>prevalence</i>	Initial prevalence (proportion)
<i>contact_rate</i>	Average number of contacts (interactions) per step.
<i>prob_transmission</i>	Probability of transmission
<i>prob_recovery</i>	Probability of recovery

16.51.3 Member Function Documentation

16.51.3.1 clone_ptr()

```
template<typename TSeq >
Model< TSeq > * ModelSIRLogit< TSeq >::clone_ptr [inline], [virtual]
```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

<i>copy</i>	
-------------	--

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.51.3.2 reset()

```
template<typename TSeq >
void ModelSIRLogit< TSeq >::reset [inline], [virtual]
```

Reset the model.

Resetting the model will:

- clear the database
- restore the population (if `set_backup()` was called before)
- re-distribute tools
- re-distribute viruses
- set the date to 0

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

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

- `epiworld.hpp`

16.52 ModelSIRLogit< TSeq > Class Template Reference

Template for a Susceptible-Infected-Removed (SIR) model.

```
#include <sirlogit.hpp>
```

Inheritance diagram for ModelSIRLogit< TSeq >:



Collaboration diagram for ModelSIRLogit< TSeq >:



Public Member Functions

- [ModelSIRLogit](#) ([ModelSIRLogit](#)< TSeq > &model, const std::string &vname, double *data, size_t ncols, std::vector< double > coefs_infect, std::vector< double > coefs_recover, std::vector< size_t > coef_infect_cols, std::vector< size_t > coef_recover_cols, epiworld_double transmission_rate, epiworld_double recovery_rate, epiworld_double prevalence)

Template for a Susceptible-Infected-Removed (SIR) model.

- **ModelSIRLogit** (const std::string &vname, double *data, size_t ncols, std::vector< double > coefs_infect, std::vector< double > coefs_recover, std::vector< size_t > coef_infect_cols, std::vector< size_t > coef_recover_cols, epiworld_double transmission_rate, epiworld_double recovery_rate, epiworld_double prevalence)
- [ModelSIRLogit](#)< TSeq > & [run](#) (epiworld_fast_uint ndays, int seed=-1)
Runs the simulation (after initialization)
- [Model](#)< TSeq > * [clone_ptr](#) ()
Advanced usage: Makes a copy of data and returns it as undeleted pointer.
- void [reset](#) ()
Reset the model.

Public Attributes

- `std::vector< double > coefs_infect`
- `std::vector< double > coefs_recover`
- `std::vector< size_t > coef_infect_cols`
- `std::vector< size_t > coef_recover_cols`

Additional Inherited Members

16.52.1 Detailed Description

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
class ModelSIRLogit< TSeq >
```

Template for a Susceptible-Infected-Removed (SIR) model.

In this model, infection and recovery probabilities are computed using a logit model. Particularly, the probability of infection is computed as:

$$\frac{1}{1 + \exp(-(\beta_0 E_i + \sum_{i=1}^n \beta_i x_i))}$$

where β_0 is the exposure coefficient and E_i is the exposure number, β_i are the coefficients for the features x_i of the agents, and n is the number of features. The probability of recovery is computed as:

$$\frac{1}{1 + \exp(-(\sum_{i=1}^n \beta_i x_i))}$$

where β_i are the coefficients for the features x_i of the agents, and n is the number of features.

Parameters

<i>TSeq</i>	Type of the sequence (e.g. <code>std::vector</code> , <code>std::deque</code>)
-------------	---

16.52.2 Constructor & Destructor Documentation

16.52.2.1 ModelSIRLogit()

```
template<typename TSeq >
ModelSIRLogit< TSeq >::ModelSIRLogit (
    ModelSIRLogit< TSeq > & model,
    const std::string & vname,
    double * data,
    size_t ncols,
```

```

std::vector< double > coefs_infect,
std::vector< double > coefs_recover,
std::vector< size_t > coef_infect_cols,
std::vector< size_t > coef_recover_cols,
epiworld_double transmission_rate,
epiworld_double recovery_rate,
epiworld_double prevalence ) [inline]

```

Template for a Susceptible-Infected-Removed (SIR) model.

Parameters

<i>vname</i>	Name of the virus.
<i>coefs_infect</i>	Double ptr. Infection coefficients.
<i>coefs_recover</i>	Double ptr. Recovery coefficients.
<i>nccoef_infect</i>	Unsigned int. Number of infection coefficients.
<i>nccoef_recover</i>	Unsigned int. Number of recovery coefficients.
<i>coef_infect_cols</i>	Vector<unsigned int>. Ids of infection vars.
<i>coef_recover_cols</i>	Vector<unsigned int>. Ids of recover vars.
<i>model</i>	A Model<TSeq> object where to set up the SIR.
<i>vname</i>	std::string Name of the virus
<i>prevalence</i>	Initial prevalence (proportion)
<i>contact_rate</i>	Average number of contacts (interactions) per step.
<i>prob_transmission</i>	Probability of transmission
<i>prob_recovery</i>	Probability of recovery

16.52.3 Member Function Documentation

16.52.3.1 clone_ptr()

```

template<typename TSeq >
Model< TSeq > * ModelSIRLogit< TSeq >::clone_ptr [inline], [virtual]

```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

<i>copy</i>	
-------------	--

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.52.3.2 reset()

```

template<typename TSeq >
void ModelSIRLogit< TSeq >::reset [inline], [virtual]

```

Reset the model.

Resetting the model will:

- clear the database
- restore the population (if `set_backup()` was called before)
- re-distribute tools
- re-distribute viruses
- set the date to 0

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

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

- `include/epiworld/models/sirlogit.hpp`

16.53 epiworld::epimodels::ModelSIRMixing< TSeq > Class Template Reference

Inheritance diagram for `epiworld::epimodels::ModelSIRMixing< TSeq >`:



Collaboration diagram for `epiworld::epimodels::ModelSIRMixing< TSeq >`:



Public Member Functions

- `ModelSIRMixing` (`ModelSIRMixing< TSeq > &model`, `const std::string &vname`, `epiworld_fast_uint n`, `epiworld_double prevalence`, `epiworld_double contact_rate`, `epiworld_double transmission_rate`, `epiworld_double recovery_rate`, `std::vector< double > contact_matrix`)
Constructs a `ModelSIRMixing` object.
- `ModelSIRMixing` (`const std::string &vname`, `epiworld_fast_uint n`, `epiworld_double prevalence`, `epiworld_double contact_rate`, `epiworld_double transmission_rate`, `epiworld_double recovery_rate`, `std::vector< double > contact_matrix`)
Constructs a `ModelSIRMixing` object.
- `ModelSIRMixing< TSeq > &run` (`epiworld_fast_uint ndays`, `int seed=-1`)
Runs the simulation (after initialization)
- `void reset` ()
Reset the model.
- `Model< TSeq > * clone_ptr` ()
Advanced usage: Makes a copy of data and returns it as undeleted pointer.
- `ModelSIRMixing< TSeq > &initial_states` (`std::vector< double > proportions_`, `std::vector< int > queue_`, `_={}`)
Set the initial states of the model.
- `size_t get_n_infected` (`size_t group`) `const`
- `void set_contact_matrix` (`std::vector< double > cmat`)

Static Public Attributes

- `static const int SUSCEPTIBLE` = 0
- `static const int INFECTED` = 1
- `static const int RECOVERED` = 2

Additional Inherited Members

16.53.1 Constructor & Destructor Documentation

16.53.1.1 `ModelSIRMixing()` [1/2]

```
template<typename TSeq >
ModelSIRMixing< TSeq >::ModelSIRMixing (
    ModelSIRMixing< TSeq > & model,
    const std::string & vname,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double recovery_rate,
    std::vector< double > contact_matrix ) [inline]
```

Constructs a `ModelSIRMixing` object.

Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.

Parameters

<i>model</i>	A reference to an existing ModelSIRMixing object.
<i>vname</i>	The name of the ModelSIRMixing object.
<i>n</i>	The number of entities in the model.
<i>prevalence</i>	The initial prevalence of the disease in the model.
<i>contact_rate</i>	The contact rate between entities in the model.
<i>transmission_rate</i>	The transmission rate of the disease in the model.
<i>recovery_rate</i>	The recovery rate of the disease in the model.
<i>contact_matrix</i>	The contact matrix between entities in the model.
<i>model</i>	A Model<TSeq> object where to set up the SIR.
<i>vname</i>	std::string Name of the virus
<i>prevalence</i>	Initial prevalence (proportion)
<i>contact_rate</i>	Average number of contacts (interactions) per step.
<i>transmission_rate</i>	Probability of transmission
<i>recovery_rate</i>	Probability of recovery

16.53.1.2 ModelSIRMixing() [2/2]

```
template<typename TSeq >
ModelSIRMixing< TSeq >::ModelSIRMixing (
    const std::string & vname,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double recovery_rate,
    std::vector< double > contact_matrix ) [inline]
```

Constructs a [ModelSIRMixing](#) object.

Parameters

<i>vname</i>	The name of the ModelSIRMixing object.
<i>n</i>	The number of entities in the model.
<i>prevalence</i>	The initial prevalence of the disease in the model.
<i>contact_rate</i>	The contact rate between entities in the model.
<i>transmission_rate</i>	The transmission rate of the disease in the model.
<i>recovery_rate</i>	The recovery rate of the disease in the model.
<i>contact_matrix</i>	The contact matrix between entities in the model.

16.53.2 Member Function Documentation

16.53.2.1 `clone_ptr()`

```
template<typename TSeq >
Model< TSeq > * ModelSIRMixing< TSeq >::clone_ptr [inline], [virtual]
```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

<i>copy</i>	
-------------	--

Reimplemented from `epiworld::Model< EPI_DEFAULT_TSEQ >`.

16.53.2.2 `initial_states()`

```
template<typename TSeq >
ModelSIRMixing< TSeq > & ModelSIRMixing< TSeq >::initial_states (
    std::vector< double > proportions_,
    std::vector< int > queue_ = {} ) [inline], [virtual]
```

Set the initial states of the model.

Parameters

<i>proportions_</i> ↩	Double vector with a single element:
—	<ul style="list-style-type: none"> The proportion of non-infected individuals who have recovered.

Reimplemented from `epiworld::Model< EPI_DEFAULT_TSEQ >`.

16.53.2.3 `reset()`

```
template<typename TSeq >
void ModelSIRMixing< TSeq >::reset [inline], [virtual]
```

Reset the model.

Resetting the model will:

- clear the database
- restore the population (if `set_backup()` was called before)
- re-distribute tools
- re-distribute viruses
- set the date to 0

Reimplemented from `epiworld::Model< EPI_DEFAULT_TSEQ >`.

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

- `epiworld.hpp`

16.54 ModelSIRMixing< TSeq > Class Template Reference

Inheritance diagram for ModelSIRMixing< TSeq >:



Collaboration diagram for ModelSIRMixing< TSeq >:



Public Member Functions

- [ModelSIRMixing](#) ([ModelSIRMixing](#)< TSeq > &model, const std::string &vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double recovery_rate, std::vector< double > contact_matrix)
Constructs a [ModelSIRMixing](#) object.
- [ModelSIRMixing](#) (const std::string &vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double recovery_rate, std::vector< double > contact_matrix)
Constructs a [ModelSIRMixing](#) object.
- [ModelSIRMixing](#)< TSeq > & [run](#) (epiworld_fast_uint ndays, int seed=-1)
Runs the simulation (after initialization)
- void [reset](#) ()
Reset the model.
- [Model](#)< TSeq > * [clone_ptr](#) ()
Advanced usage: Makes a copy of data and returns it as undeleted pointer.

- [ModelSIRMixing](#)< TSeq > & [initial_states](#) (std::vector< double > proportions_, std::vector< int > queue_ ←
_={})
Set the initial states of the model.
- size_t **get_n_infected** (size_t group) const
- void **set_contact_matrix** (std::vector< double > cmat)

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **INFECTED** = 1
- static const int **RECOVERED** = 2

Additional Inherited Members

16.54.1 Constructor & Destructor Documentation

16.54.1.1 ModelSIRMixing() [1/2]

```
template<typename TSeq >
ModelSIRMixing< TSeq >::ModelSIRMixing (
    ModelSIRMixing< TSeq > & model,
    const std::string & vname,
    epiworld_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double recovery_rate,
    std::vector< double > contact_matrix ) [inline]
```

Constructs a [ModelSIRMixing](#) object.

Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.

Parameters

<i>model</i>	A reference to an existing ModelSIRMixing object.
<i>vname</i>	The name of the ModelSIRMixing object.
<i>n</i>	The number of entities in the model.
<i>prevalence</i>	The initial prevalence of the disease in the model.
<i>contact_rate</i>	The contact rate between entities in the model.
<i>transmission_rate</i>	The transmission rate of the disease in the model.
<i>recovery_rate</i>	The recovery rate of the disease in the model.
<i>contact_matrix</i>	The contact matrix between entities in the model.
<i>model</i>	A Model<TSeq> object where to set up the SIR.
<i>vname</i>	std::string Name of the virus
<i>prevalence</i>	Initial prevalence (proportion)
<i>contact_rate</i>	Average number of contacts (interactions) per step.
<i>transmission_rate</i>	Probability of transmission
<i>recovery_rate</i>	Probability of recovery

16.54.1.2 ModelSIRMixing() [2/2]

```
template<typename TSeq >
ModelSIRMixing< TSeq >::ModelSIRMixing (
    const std::string & vname,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double recovery_rate,
    std::vector< double > contact_matrix ) [inline]
```

Constructs a [ModelSIRMixing](#) object.

Parameters

<i>vname</i>	The name of the ModelSIRMixing object.
<i>n</i>	The number of entities in the model.
<i>prevalence</i>	The initial prevalence of the disease in the model.
<i>contact_rate</i>	The contact rate between entities in the model.
<i>transmission_rate</i>	The transmission rate of the disease in the model.
<i>recovery_rate</i>	The recovery rate of the disease in the model.
<i>contact_matrix</i>	The contact matrix between entities in the model.

16.54.2 Member Function Documentation

16.54.2.1 clone_ptr()

```
template<typename TSeq >
Model< TSeq > * ModelSIRMixing< TSeq >::clone_ptr [inline], [virtual]
```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

<i>copy</i>	
-------------	--

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.54.2.2 initial_states()

```
template<typename TSeq >
ModelSIRMixing< TSeq > & ModelSIRMixing< TSeq >::initial_states (
```

```
std::vector< double > proportions_,
std::vector< int > queue_ = {} ) [inline], [virtual]
```

Set the initial states of the model.

Parameters

<i>proportions_</i> ↔	Double vector with a single element:
—	<ul style="list-style-type: none"> The proportion of non-infected individuals who have recovered.

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

16.54.2.3 reset()

```
template<typename TSeq >
void ModelSIRMixing< TSeq >::reset [inline], [virtual]
```

Reset the model.

Resetting the model will:

- clear the database
- restore the population (if `set_backup()` was called before)
- re-distribute tools
- re-distribute viruses
- set the date to 0

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

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

- `include/epiworld/models/sirmixing.hpp`

16.55 epiworld::epimodels::ModelSIS< TSeq > Class Template Reference

Template for a Susceptible-Infected-Susceptible (SIS) model.

```
#include <epiworld.hpp>
```

Inheritance diagram for `epiworld::epimodels::ModelSIS< TSeq >`:



Collaboration diagram for `epiworld::epimodels::ModelSIS< TSeq >`:



Public Member Functions

- **ModelSIS** ([ModelSIS](#)< TSeq > &model, const std::string &vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double recovery_rate)
- **ModelSIS** (const std::string &vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double recovery_rate)

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **INFECTED** = 1

Additional Inherited Members

16.55.1 Detailed Description

```
template<typename TSeq = EPI_DEFAULT_TSEQ>  
class epiworld::epimodels::ModelSIS< TSeq >
```

Template for a Susceptible-Infected-Susceptible (SIS) model.

Parameters

<i>vname</i>	std::string Name of the virus
<i>initial_prevalence</i>	epiworld_double Initial prevalence
<i>initial_efficacy</i>	epiworld_double Initial susceptibility_reduction of the immune system
<i>initial_recovery</i>	epiworld_double Initial recovery_rate rate of the immune system

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

- epiworld.hpp

16.56 ModelSIS< TSeq > Class Template Reference

Template for a Susceptible-Infected-Susceptible (SIS) model.

```
#include <sis.hpp>
```

Inheritance diagram for ModelSIS< TSeq >:



Collaboration diagram for ModelSIS< TSeq >:



Public Member Functions

- **ModelSIS** ([ModelSIS](#)< TSeq > &model, const std::string &vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double recovery_rate)
- **ModelSIS** (const std::string &vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double recovery_rate)

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **INFECTED** = 1

Additional Inherited Members

16.56.1 Detailed Description

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
class ModelSIS< TSeq >
```

Template for a Susceptible-Infected-Susceptible (SIS) model.

Parameters

<i>vname</i>	std::string Name of the virus
<i>initial_prevalence</i>	epiworld_double Initial prevalence
<i>initial_efficacy</i>	epiworld_double Initial susceptibility_reduction of the immune system
<i>initial_recovery</i>	epiworld_double Initial recovery_rate rate of the immune system

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

- `include/epiworld/models/sis.hpp`

16.57 `epiworld::epimodels::ModelSISD< TSeq >` Class Template Reference

Template for a Susceptible-Infected-Susceptible-Deceased (SISD) model.

```
#include <epiworld.hpp>
```

Inheritance diagram for `epiworld::epimodels::ModelSISD< TSeq >`:



Collaboration diagram for `epiworld::epimodels::ModelSISD< TSeq >`:



Public Member Functions

- **ModelSISD** ([ModelSISD](#)< TSeq > &model, const std::string &vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double recovery_rate, epiworld_double death_rate)
- **ModelSISD** (const std::string &vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double recovery_rate, epiworld_double death_rate)

Additional Inherited Members

16.57.1 Detailed Description

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
class epiworld::epimodels::ModelSISD< TSeq >
```

Template for a Susceptible-Infected-Susceptible-Deceased (SISD) model.

Parameters

<i>vname</i>	std::string Name of the virus
<i>initial_prevalence</i>	epiworld_double Initial prevalence
<i>initial_efficacy</i>	epiworld_double Initial susceptibility_reduction of the immune system
<i>initial_recovery</i>	epiworld_double Initial recovery_rate rate of the immune system
<i>inital_death</i>	epiworld_double Initial death_rate of the immune system

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

- epiworld.hpp

16.58 ModelSISD< TSeq > Class Template Reference

Template for a Susceptible-Infected-Susceptible-Deceased (SISD) model.

```
#include <sisd.hpp>
```

Inheritance diagram for ModelSISD< TSeq >:



Collaboration diagram for ModelSISD< TSeq >:



Public Member Functions

- **ModelSISD** ([ModelSISD](#)< TSeq > &model, const std::string &vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double recovery_rate, epiworld_double death_rate)
- **ModelSISD** (const std::string &vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double recovery_rate, epiworld_double death_rate)

Additional Inherited Members

16.58.1 Detailed Description

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
class ModelSISD< TSeq >
```

Template for a Susceptible-Infected-Susceptible-Deceased (SISD) model.

Parameters

<i>vname</i>	std::string Name of the virus
<i>initial_prevalence</i>	epiworld_double Initial prevalence
<i>initial_efficiency</i>	epiworld_double Initial susceptibility_reduction of the immune system
<i>initial_recovery</i>	epiworld_double Initial recovery_rate rate of the immune system
<i>initial_death</i>	epiworld_double Initial death_rate of the immune system

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

- include/epiworld/models/sisd.hpp

16.59 epiworld::epimodels::ModelSURV< TSeq > Class Template Reference

Inheritance diagram for epiworld::epimodels::ModelSURV< TSeq >:



Collaboration diagram for epiworld::epimodels::ModelSURV< TSeq >:



Public Member Functions

- void `reset` ()
Reset the model.

Construct a new ModelSURV object

The `ModelSURV` class simulates a surveillance model where agents can be isolated, even if asymptomatic.

Parameters

<code>vname</code>	<i>String. Name of the virus</i>
<code>prevalence</code>	<i>Integer. Number of initial cases of the virus.</i>
<code>efficacy_vax</code>	<i>Double. Efficacy of the vaccine (1 - P(acquire the disease)).</i>
<code>latent_period</code>	<i>Double. Shape parameter of a Gamma (latent_period, 1) distribution. This coincides with the expected number of latent days.</i>
<code>infect_period</code>	<i>Double. Shape parameter of a Gamma (infected_period, 1) distribution. This coincides with the expected number of infectious days.</i>
<code>prob_symptoms</code>	<i>Double. Probability of generating symptoms.</i>
<code>prop_vaccinated</code>	<i>Double. Probability of vaccination. Coincides with the initial prevalence of vaccinated individuals.</i>
<code>prop_vax_redux_transm</code>	<i>Double. Factor by which the vaccine reduces transmissibility.</i>
<code>prop_vax_redux_infect</code>	<i>Double. Factor by which the vaccine reduces the chances of becoming infected.</i>
<code>surveillance_prob</code>	<i>Double. Probability of testing an agent.</i>
<code>prob_transmission</code>	<i>Double. Raw transmission probability.</i>
<code>prob_death</code>	<i>Double. Raw probability of death for symptomatic individuals.</i>
<code>prob_noreinfect</code>	<i>Double. Probability of no re-infection.</i>

This model features the following states:

- *Susceptible*
- *Latent*
- *Symptomatic*
- *Symptomatic isolated*

- *Asymptomatic*
- *Asymptomatic isolated*
- *Recovered*
- *Removed*

Returns

An object of class *epiworld_surv*

- **ModelSURV** ()
- **ModelSURV** ([ModelSURV](#)< TSeq > &model, const std::string &vname, epiworld_fast_uint prevalence=50, epiworld_double efficacy_vax=0.9, epiworld_double latent_period=3u, epiworld_double infect_↵_period=6u, epiworld_double prob_symptoms=0.6, epiworld_double prop_vaccinated=0.25, epiworld_↵_double prop_vax_redux_transm=0.5, epiworld_double prop_vax_redux_infect=0.5, epiworld_double surveillance_prob=0.001, epiworld_double prob_transmission=1.0, epiworld_double prob_death=0.001, epiworld_double prob_noreinfect=0.9)
- **ModelSURV** (const std::string &vname, epiworld_fast_uint prevalence=50, epiworld_double efficacy_↵_vax=0.9, epiworld_double latent_period=3u, epiworld_double infect_period=6u, epiworld_double prob_↵_symptoms=0.6, epiworld_double prop_vaccinated=0.25, epiworld_double prop_vax_redux_transm=0.5, epiworld_double prop_vax_redux_infect=0.5, epiworld_double surveillance_prob=0.001, epiworld_double prob_transmission=1.0, epiworld_double prob_death=0.001, epiworld_double prob_noreinfect=0.9)

Public Attributes

- std::vector< epiworld_double > [days_latent_and_infectious](#)
Vector of days spent in latent and infectious states A row-major matrix.

Additional Inherited Members

16.59.1 Member Function Documentation

16.59.1.1 reset()

```
template<typename TSeq >
void ModelSURV< TSeq >::reset [inline], [virtual]
```

Reset the model.

Resetting the model will:

- clear the database
- restore the population (if `set_backup()` was called before)
- re-distribute tools
- re-distribute viruses
- set the date to 0

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

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

- epiworld.hpp

16.60 ModelSURV< TSeq > Class Template Reference

Inheritance diagram for ModelSURV< TSeq >:



Collaboration diagram for ModelSURV< TSeq >:



Public Member Functions

- void [reset](#) ()
Reset the model.

Construct a new ModelSURV object

The [ModelSURV](#) class simulates a surveillance model where agents can be isolated, even if asymptomatic.

Parameters

vname	<i>String. Name of the virus</i>
prevalence	<i>Integer. Number of initial cases of the virus.</i>
efficacy_vax	<i>Double. Efficacy of the vaccine (1 - P(acquire the disease)).</i>
latent_period	<i>Double. Shape parameter of a Gamma (latent_period, 1) distribution. This coincides with the expected number of latent days.</i>

Parameters

infect_period	<i>Double. Shape parameter of a Gamma (infected_period, 1) distribution. This coincides with the expected number of infectious days.</i>
prob_symptoms	<i>Double. Probability of generating symptoms.</i>
prop_vaccinated	<i>Double. Probability of vaccination. Coincides with the initial prevalence of vaccinated individuals.</i>
prop_vax_redux_transm	<i>Double. Factor by which the vaccine reduces transmissibility.</i>
prop_vax_redux_infect	<i>Double. Factor by which the vaccine reduces the chances of becoming infected.</i>
surveillance_prob	<i>Double. Probability of testing an agent.</i>
prob_transmission	<i>Double. Raw transmission probability.</i>
prob_death	<i>Double. Raw probability of death for symptomatic individuals.</i>
prob_noreinfect	<i>Double. Probability of no re-infection.</i>

This model features the following states:

- *Susceptible*
- *Latent*
- *Symptomatic*
- *Symptomatic isolated*
- *Asymptomatic*
- *Asymptomatic isolated*
- *Recovered*
- *Removed*

Returns

An object of class `epiworld_surv`

- **ModelSURV** ()
- **ModelSURV** ([ModelSURV](#) < TSeq > &model, const std::string &vname, epiworld_fast_uint prevalence=50, epiworld_double efficacy_vax=0.9, epiworld_double latent_period=3u, epiworld_double infect_↵_period=6u, epiworld_double prob_symptoms=0.6, epiworld_double prop_vaccinated=0.25, epiworld_↵_double prop_vax_redux_transm=0.5, epiworld_double prop_vax_redux_infect=0.5, epiworld_double surveillance_prob=0.001, epiworld_double prob_transmission=1.0, epiworld_double prob_death=0.001, epiworld_double prob_noreinfect=0.9)
- **ModelSURV** (const std::string &vname, epiworld_fast_uint prevalence=50, epiworld_double efficacy_↵_vax=0.9, epiworld_double latent_period=3u, epiworld_double infect_period=6u, epiworld_double prob_↵_symptoms=0.6, epiworld_double prop_vaccinated=0.25, epiworld_double prop_vax_redux_transm=0.5, epiworld_double prop_vax_redux_infect=0.5, epiworld_double surveillance_prob=0.001, epiworld_double prob_transmission=1.0, epiworld_double prob_death=0.001, epiworld_double prob_noreinfect=0.9)

Public Attributes

- std::vector< epiworld_double > [days_latent_and_infectious](#)
Vector of days spent in latent and infectious states A row-major matrix.

Additional Inherited Members**16.60.1 Member Function Documentation**

16.60.1.1 reset()

```
template<typename TSeq >
void ModelSURV< TSeq >::reset [inline], [virtual]
```

Reset the model.

Resetting the model will:

- clear the database
- restore the population (if `set_backup()` was called before)
- re-distribute tools
- re-distribute viruses
- set the date to 0

Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

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

- `include/epiworld/models/surveillance.hpp`

16.61 Network< Nettype, Nodetype, Edgetype > Class Template Reference

Public Member Functions

- **NType** ()
- Edgetype **operator()** (int i, int j)
- bool **is_directed** () const
- size_t **vcount** () const
- size_t **ecount** () const
- void **add_edge** (int i, int j)
- void **rm_edge** (int i, int j)

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

- `include/epiworld/network-bones.hpp`

16.62 epiworld::PersonTools< TSeq > Class Template Reference

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

- `epiworld.hpp`

16.63 PersonTools< TSeq > Class Template Reference

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

- include/epiworld/config.hpp

16.64 epiworld::Progress Class Reference

A simple progress bar.

```
#include <epiworld.hpp>
```

Public Member Functions

- **Progress** (int n_, int width_)
- void **start** ()
- void **next** ()
- void **end** ()

16.64.1 Detailed Description

A simple progress bar.

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

- epiworld.hpp

16.65 Progress Class Reference

A simple progress bar.

```
#include <progress.hpp>
```

Public Member Functions

- **Progress** (int n_, int width_)
- void **start** ()
- void **next** ()
- void **end** ()

16.65.1 Detailed Description

A simple progress bar.

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

- include/epiworld/progress.hpp

16.66 epiworld::Queue< TSeq > Class Template Reference

Controls which agents are verified at each step.

```
#include <epiworld.hpp>
```

Public Member Functions

- void **operator+=** ([Agent](#)< TSeq > *p)
- void **operator-=** ([Agent](#)< TSeq > *p)
- epiworld_fast_int & **operator[]** (epiworld_fast_uint i)
- void **reset** ()
- bool **operator==** (const [Queue](#)< TSeq > &other) const
- bool **operator!=** (const [Queue](#)< TSeq > &other) const

Static Public Attributes

- static const int **NoOne** = 0
- static const int **OnlySelf** = 1
- static const int **Everyone** = 2

Friends

- class **Model**< TSeq >

16.66.1 Detailed Description

```
template<typename TSeq>
class epiworld::Queue< TSeq >
```

Controls which agents are verified at each step.

The idea is that only agents who are either in an infected state or have an infected neighbor should be checked. Otherwise it makes no sense (no chance to recover or capture the disease).

Template Parameters

<i>TSeq</i>	
-------------	--

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

- epiworld.hpp

16.67 Queue< TSeq > Class Template Reference

Controls which agents are verified at each step.

```
#include <queue-bones.hpp>
```

Public Member Functions

- void **operator+=** ([Agent](#)< TSeq > *p)
- void **operator-=** ([Agent](#)< TSeq > *p)
- [epiworld_fast_int](#) & **operator[]** ([epiworld_fast_uint](#) i)
- void **reset** ()
- bool **operator==** (const [Queue](#)< TSeq > &other) const
- bool **operator!=** (const [Queue](#)< TSeq > &other) const

Static Public Attributes

- static const int **NoOne** = 0
- static const int **OnlySelf** = 1
- static const int **Everyone** = 2

Friends

- class **Model**< TSeq >

16.67.1 Detailed Description

```
template<typename TSeq>
class Queue< TSeq >
```

Controls which agents are verified at each step.

The idea is that only agents who are either in an infected state or have an infected neighbor should be checked. Otherwise it makes no sense (no chance to recover or capture the disease).

Template Parameters

<i>TSeq</i>	
-------------	--

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

- include/epiworld/agent-bones.hpp
- include/epiworld/queue-bones.hpp

16.68 RandGraph Class Reference

Public Member Functions

- **RandGraph** (int N_)
- void **init** (int s)
- void **set_rand_engine** (std::shared_ptr< std::mt19937 > &e)
- [epiworld_double](#) **runif** ()

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

- include/epiworld/random_graph.hpp

16.69 epiworld::SAMPLETYPE Class Reference

Static Public Attributes

- static const int **MODEL** = 0
- static const int **ENTITY** = 1
- static const int **AGENT** = 2

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

- epiworld.hpp

16.70 SAMPLETYPE Class Reference

Static Public Attributes

- static const int **MODEL** = 0
- static const int **ENTITY** = 1
- static const int **AGENT** = 2

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

- include/epiworld/agentssample-bones.hpp

16.71 epiworld::Tool< TSeq > Class Template Reference

[Tools](#) for defending the agent against the virus.

```
#include <epiworld.hpp>
```

Public Member Functions

- **Tool** (std::string name="unknown tool")
- **Tool** (std::string name, epiworld_double prevalence, bool as_proportion)
- void **set_sequence** (TSeq d)
- void **set_sequence** (std::shared_ptr< TSeq > d)
- **EPI_TYPENAME_TRAITS** (TSeq, int) get_sequence()
- void **set_name** (std::string name)
- std::string **get_name** () const
- [Agent](#)< TSeq > * **get_agent** ()
- int **get_id** () const
- void **set_id** (int id)
- void **set_date** (int d)
- int **get_date** () const
- void **set_state** (epiworld_fast_int init, epiworld_fast_int post)
- void **set_queue** (epiworld_fast_int init, epiworld_fast_int post)
- void **get_state** (epiworld_fast_int *init, epiworld_fast_int *post)
- void **get_queue** (epiworld_fast_int *init, epiworld_fast_int *post)
- bool **operator==** (const [Tool](#)< TSeq > &other) const
- bool **operator!=** (const [Tool](#)< TSeq > &other) const
- void **print** () const
- void **distribute** ([Model](#)< TSeq > *model)
- void **set_distribution** (ToolToAgentFun< TSeq > fun)

Get and set the tool functions

Parameters

v	<i>The virus over which to operate</i>
fun	<i>the function to be used</i>

Returns

epiworld_double

- epiworld_double **get_susceptibility_reduction** (VirusPtr< TSeq > v, [Model](#)< TSeq > *model)
- epiworld_double **get_transmission_reduction** (VirusPtr< TSeq > v, [Model](#)< TSeq > *model)
- epiworld_double **get_recovery_enhancer** (VirusPtr< TSeq > v, [Model](#)< TSeq > *model)
- epiworld_double **get_death_reduction** (VirusPtr< TSeq > v, [Model](#)< TSeq > *model)
- void **set_susceptibility_reduction_fun** (ToolFun< TSeq > fun)
- void **set_transmission_reduction_fun** (ToolFun< TSeq > fun)
- void **set_recovery_enhancer_fun** (ToolFun< TSeq > fun)
- void **set_death_reduction_fun** (ToolFun< TSeq > fun)
- void **set_susceptibility_reduction** (epiworld_double *prob)
- void **set_transmission_reduction** (epiworld_double *prob)
- void **set_recovery_enhancer** (epiworld_double *prob)
- void **set_death_reduction** (epiworld_double *prob)
- void **set_susceptibility_reduction** (epiworld_double prob)
- void **set_transmission_reduction** (epiworld_double prob)
- void **set_recovery_enhancer** (epiworld_double prob)
- void **set_death_reduction** (epiworld_double prob)

Friends

- class **Agent**< TSeq >
- class **Model**< TSeq >
- void **default_add_tool** ([Event](#)< TSeq > &a, [Model](#)< TSeq > *m)
- void **default_rm_tool** ([Event](#)< TSeq > &a, [Model](#)< TSeq > *m)

16.71.1 Detailed Description

```
template<typename TSeq>
class epiworld::Tool< TSeq >
```

[Tools](#) for defending the agent against the virus.

Template Parameters

<i>TSeq</i>	Type of sequence
-------------	------------------

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

- epiworld.hpp

16.72 Tool< TSeq > Class Template Reference

[Tools](#) for defending the agent against the virus.

```
#include <tool-bones.hpp>
```

Public Member Functions

- **Tool** (std::string name="unknown tool")
- **Tool** (std::string name, epiworld_double prevalence, bool as_proportion)
- void **set_sequence** (TSeq d)
- void **set_sequence** (std::shared_ptr< TSeq > d)
- **EPI_TYPENAME_TRAITS** (TSeq, int) get_sequence()
- void **set_name** (std::string name)
- std::string **get_name** () const
- [Agent](#)< TSeq > * **get_agent** ()
- int **get_id** () const
- void **set_id** (int id)
- void **set_date** (int d)
- int **get_date** () const
- void **set_state** (epiworld_fast_int init, epiworld_fast_int post)
- void **set_queue** (epiworld_fast_int init, epiworld_fast_int post)
- void **get_state** (epiworld_fast_int *init, epiworld_fast_int *post)
- void **get_queue** (epiworld_fast_int *init, epiworld_fast_int *post)
- bool **operator==** (const [Tool](#)< TSeq > &other) const
- bool **operator!=** (const [Tool](#)< TSeq > &other) const
- void **print** () const
- void **distribute** ([Model](#)< TSeq > *model)
- void **set_distribution** (ToolToAgentFun< TSeq > fun)
- void **set_sequence** (int d)
- bool **operator==** (const [Tool](#)< std::vector< int >> &other) const
- void **set_sequence** (int d)
- bool **operator==** (const [Tool](#)< std::vector< int >> &other) const

Get and set the tool functions

Parameters

v	<i>The virus over which to operate</i>
fun	<i>the function to be used</i>

Returns

epiworld_double

- epiworld_double **get_susceptibility_reduction** (VirusPtr< TSeq > v, [Model](#)< TSeq > *model)
- epiworld_double **get_transmission_reduction** (VirusPtr< TSeq > v, [Model](#)< TSeq > *model)
- epiworld_double **get_recovery_enhancer** (VirusPtr< TSeq > v, [Model](#)< TSeq > *model)
- epiworld_double **get_death_reduction** (VirusPtr< TSeq > v, [Model](#)< TSeq > *model)
- void **set_susceptibility_reduction_fun** (ToolFun< TSeq > fun)
- void **set_transmission_reduction_fun** (ToolFun< TSeq > fun)
- void **set_recovery_enhancer_fun** (ToolFun< TSeq > fun)
- void **set_death_reduction_fun** (ToolFun< TSeq > fun)
- void **set_susceptibility_reduction** (epiworld_double *prob)
- void **set_transmission_reduction** (epiworld_double *prob)
- void **set_recovery_enhancer** (epiworld_double *prob)
- void **set_death_reduction** (epiworld_double *prob)
- void **set_susceptibility_reduction** (epiworld_double prob)
- void **set_transmission_reduction** (epiworld_double prob)
- void **set_recovery_enhancer** (epiworld_double prob)
- void **set_death_reduction** (epiworld_double prob)

Friends

- class **Agent**< TSeq >
- class **Model**< TSeq >
- void **default_add_tool** ([Event](#)< TSeq > &a, [Model](#)< TSeq > *m)
- void **default_rm_tool** ([Event](#)< TSeq > &a, [Model](#)< TSeq > *m)

16.72.1 Detailed Description

```
template<typename TSeq>
class Tool< TSeq >
```

[Tools](#) for defending the agent against the virus.

Template Parameters

<i>TSeq</i>	Type of sequence
-------------	------------------

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

- include/epiworld/agent-bones.hpp
- include/epiworld/tool-bones.hpp
- include/epiworld/tool-meat.hpp

16.73 epiworld::ToolFunctions< TSeq > Class Template Reference

Helper class to store the functions avoiding multiple shared_pointers (we have only one for the four of these)

```
#include <epiworld.hpp>
```

Public Attributes

- ToolFun< TSeq > **susceptibility_reduction** = nullptr
- ToolFun< TSeq > **transmission_reduction** = nullptr
- ToolFun< TSeq > **recovery_enhancer** = nullptr
- ToolFun< TSeq > **death_reduction** = nullptr
- ToolToAgentFun< TSeq > **dist** = nullptr

16.73.1 Detailed Description

```
template<typename TSeq>
class epiworld::ToolFunctions< TSeq >
```

Helper class to store the functions avoiding multiple shared_pointers (we have only one for the four of these)

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

- epiworld.hpp

16.74 ToolFunctions< TSeq > Class Template Reference

Helper class to store the functions avoiding multiple shared_pointers (we have only one for the four of these)

```
#include <tool-bones.hpp>
```

Public Attributes

- ToolFun< TSeq > **susceptibility_reduction** = nullptr
- ToolFun< TSeq > **transmission_reduction** = nullptr
- ToolFun< TSeq > **recovery_enhancer** = nullptr
- ToolFun< TSeq > **death_reduction** = nullptr
- ToolToAgentFun< TSeq > **dist** = nullptr

16.74.1 Detailed Description

```
template<typename TSeq>
class ToolFunctions< TSeq >
```

Helper class to store the functions avoiding multiple shared_pointers (we have only one for the four of these)

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

- include/epiworld/tool-bones.hpp

16.75 epiworld::Tools< TSeq > Class Template Reference

Set of tools (useful for building iterators)

```
#include <epiworld.hpp>
```

Public Member Functions

- **Tools** ([Agent](#)< TSeq > &p)
- std::vector< ToolPtr< TSeq > >::iterator **begin** ()
- std::vector< ToolPtr< TSeq > >::iterator **end** ()
- ToolPtr< TSeq > & **operator()** (size_t i)
- ToolPtr< TSeq > & **operator[]** (size_t i)
- size_t **size** () const noexcept
- void **print** () const noexcept

Friends

- class **Tool**< TSeq >
- class **Agent**< TSeq >

16.75.1 Detailed Description

```
template<typename TSeq>
class epiworld::Tools< TSeq >
```

Set of tools (useful for building iterators)

Template Parameters

<i>TSeq</i>	
-------------	--

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

- epiworld.hpp

16.76 Tools< TSeq > Class Template Reference

Set of tools (useful for building iterators)

```
#include <tools-bones.hpp>
```

Public Member Functions

- **Tools** ([Agent](#)< TSeq > &p)
- std::vector< ToolPtr< TSeq > >::iterator **begin** ()
- std::vector< ToolPtr< TSeq > >::iterator **end** ()
- ToolPtr< TSeq > & **operator()** (size_t i)
- ToolPtr< TSeq > & **operator[]** (size_t i)
- size_t **size** () const noexcept
- void **print** () const noexcept

Friends

- class **Tool**< TSeq >
- class **Agent**< TSeq >

16.76.1 Detailed Description

```
template<typename TSeq>
class Tools< TSeq >
```

Set of tools (useful for building iterators)

Template Parameters

<i>TSeq</i>	
-------------	--

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

- include/epiworld/agent-bones.hpp
- include/epiworld/tools-bones.hpp

16.77 epiworld::Tools_const< TSeq > Class Template Reference

Set of [Tools](#) (const) (useful for iterators)

```
#include <epiworld.hpp>
```

Public Member Functions

- **Tools_const** (const [Agent](#)< TSeq > &p)
- std::vector< ToolPtr< TSeq > >::const_iterator **begin** () const
- std::vector< ToolPtr< TSeq > >::const_iterator **end** () const
- const ToolPtr< TSeq > & **operator**() (size_t i)
- const ToolPtr< TSeq > & **operator**[] (size_t i)
- size_t **size** () const noexcept
- void **print** () const noexcept

Friends

- class **Tool**< TSeq >
- class **Agent**< TSeq >

16.77.1 Detailed Description

```
template<typename TSeq>
class epiworld::Tools_const< TSeq >
```

Set of [Tools](#) (const) (useful for iterators)

Template Parameters

<i>TSeq</i>	
-------------	--

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

- epiworld.hpp

16.78 Tools_const< TSeq > Class Template Reference

Set of [Tools](#) (const) (useful for iterators)

```
#include <tools-bones.hpp>
```

Public Member Functions

- **Tools_const** (const [Agent](#)< TSeq > &p)
- std::vector< ToolPtr< TSeq > >::const_iterator **begin** () const
- std::vector< ToolPtr< TSeq > >::const_iterator **end** () const
- const ToolPtr< TSeq > & **operator**() (size_t i)
- const ToolPtr< TSeq > & **operator**[] (size_t i)
- size_t **size** () const noexcept
- void **print** () const noexcept

Friends

- class **Tool**< TSeq >
- class **Agent**< TSeq >

16.78.1 Detailed Description

```
template<typename TSeq>
class Tools_const< TSeq >
```

Set of [Tools](#) (const) (useful for iterators)

Template Parameters

<i>TSeq</i>	
-------------	--

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

- include/epiworld/agent-bones.hpp
- include/epiworld/tools-bones.hpp

16.79 epiworld::UserData< TSeq > Class Template Reference

Personalized data by the user.

```
#include <epiworld.hpp>
```

Public Member Functions

- **UserData** ([Model](#)< TSeq > &m)
- **UserData** ([Model](#)< TSeq > *m)
- **UserData** (std::vector< std::string > names)
Construct a new User Data object.
- std::vector< std::string > & **get_names** ()
- std::vector< int > & **get_dates** ()
- std::vector< epiworld_double > & **get_data** ()

- void **get_all** (std::vector< std::string > *names=nullptr, std::vector< int > *date=nullptr, std::vector< epiworld_double > *data=nullptr)
- epiworld_fast_uint **nrow** () const
- epiworld_fast_uint **ncol** () const
- void **write** (std::string fn)
- void **print** () const

Append data

Parameters

x	A vector of length <code>ncol()</code> (if vector), otherwise a <code>epiworld_double</code> .
j	Index of the data point, from 0 to <code>ncol()</code> - 1.

- void **add** (std::vector< epiworld_double > x)
- void **add** (epiworld_fast_uint j, epiworld_double x)

Access data

Parameters

i	Row (0 through <code>ndays</code> - 1.)
j	Column (0 through <code>ncols()</code>).

Returns

`epiworld_double&`

- epiworld_double & **operator()** (epiworld_fast_uint i, epiworld_fast_uint j)
- epiworld_double & **operator()** (epiworld_fast_uint i, std::string name)

Friends

- class **Model**< TSeq >
- class **DataBase**< TSeq >

16.79.1 Detailed Description

```
template<typename TSeq>
class epiworld::UserData< TSeq >
```

Personalized data by the user.

Template Parameters

<i>TSeq</i>	
-------------	--

16.79.2 Constructor & Destructor Documentation

16.79.2.1 UserData()

```
template<typename TSeq >
UserData< TSeq >::UserData (
    std::vector< std::string > names ) [inline]
```

Construct a new User Data object.

Parameters

<i>names</i>	A vector of names. The length of the vector sets the number of columns to record.
--------------	---

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

- epiworld.hpp

16.80 UserData< TSeq > Class Template Reference

Personalized data by the user.

```
#include <userdata-bones.hpp>
```

Public Member Functions

- **UserData** ([Model](#)< TSeq > &m)
- **UserData** ([Model](#)< TSeq > *m)
- [UserData](#) (std::vector< std::string > names)
Construct a new User Data object.
- std::vector< std::string > & **get_names** ()
- std::vector< int > & **get_dates** ()
- std::vector< epiworld_double > & **get_data** ()
- void **get_all** (std::vector< std::string > *names=nullptr, std::vector< int > *date=nullptr, std::vector< epiworld_double > *data=nullptr)
- epiworld_fast_uint **nrow** () const
- epiworld_fast_uint **ncol** () const
- void **write** (std::string fn)
- void **print** () const

Append data

Parameters

<i>x</i>	A vector of length <i>ncol</i> () (if vector), otherwise a <i>epiworld_double</i> .
<i>j</i>	Index of the data point, from 0 to <i>ncol</i> () - 1.

- void **add** (std::vector< epiworld_double > x)
- void **add** (epiworld_fast_uint j, epiworld_double x)

Access data

Parameters

i	Row (0 through <i>ndays</i> - 1.)
j	Column (0 through <i>ncols</i> ()).

Returns

epiworld_double&

- epiworld_double & **operator()** (epiworld_fast_uint i, epiworld_fast_uint j)
- epiworld_double & **operator()** (epiworld_fast_uint i, std::string name)

Friends

- class **Model**< TSeq >
- class **DataBase**< TSeq >

16.80.1 Detailed Description

```
template<typename TSeq>
class UserData< TSeq >
```

Personalized data by the user.

Template Parameters

<i>TSeq</i>	
-------------	--

16.80.2 Constructor & Destructor Documentation

16.80.2.1 UserData()

```
template<typename TSeq >
UserData< TSeq >::UserData (
    std::vector< std::string > names ) [inline]
```

Construct a new User Data object.

Parameters

<i>names</i>	A vector of names. The length of the vector sets the number of columns to record.
--------------	---

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

- include/epiworld/database-bones.hpp
- include/epiworld/userdata-bones.hpp
- include/epiworld/userdata-meat.hpp

16.81 epiworld::vecHasher< T > Struct Template Reference

Vector hasher.

```
#include <epiworld.hpp>
```

Public Member Functions

- `std::size_t operator() (std::vector< T > const &dat) const` noexcept

16.81.1 Detailed Description

```
template<typename T>
struct epiworld::vecHasher< T >
```

Vector hasher.

Template Parameters

<i>T</i>	
----------	--

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

- epiworld.hpp

16.82 vecHasher< T > Struct Template Reference

Vector hasher.

```
#include <misc.hpp>
```

Public Member Functions

- `std::size_t operator() (std::vector< T > const &dat) const` noexcept

16.82.1 Detailed Description

```
template<typename T>
struct vecHasher< T >
```

Vector hasher.

Template Parameters

<i>T</i>	
----------	--

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

- include/epiworld/misc.hpp

16.83 epiworld::Virus< TSeq > Class Template Reference

[Virus.](#)

```
#include <epiworld.hpp>
```

Public Member Functions

- **Virus** (std::string name="unknown virus")
- **Virus** (std::string name, epiworld_double prevalence, bool as_proportion)
- void **mutate** ([Model](#)< TSeq > *model)
- void **set_mutation** (MutFun< TSeq > fun)
- **EPI_TYPENAME_TRAITS** (TSeq, int) get_sequence()
- void **set_sequence** (TSeq sequence)
- [Agent](#)< TSeq > * **get_agent** ()
- void **set_agent** ([Agent](#)< TSeq > *p)
- void **set_date** (int d)
- int **get_date** () const
- void **set_id** (int idx)
- int **get_id** () const
- void **set_name** (std::string name)
- std::string **get_name** () const
- bool **operator==** (const [Virus](#)< TSeq > &other) const
- bool **operator!=** (const [Virus](#)< TSeq > &other) const
- void **print** () const

Get and set the tool functions

Parameters

v	<i>The virus over which to operate</i>
fun	<i>the function to be used</i>

Returns

epiworld_double

- epiworld_double **get_prob_infecting** ([Model](#)< TSeq > *model)
- epiworld_double **get_prob_recovery** ([Model](#)< TSeq > *model)
- epiworld_double **get_prob_death** ([Model](#)< TSeq > *model)
- epiworld_double **get_incubation** ([Model](#)< TSeq > *model)
- void **post_recovery** ([Model](#)< TSeq > *model)

- void **set_post_recovery** (PostRecoveryFun< TSeq > fun)
- void **set_post_immunity** (epiworld_double prob)
- void **set_post_immunity** (epiworld_double *prob)
- void **set_prob_infecting_fun** (VirusFun< TSeq > fun)
- void **set_prob_recovery_fun** (VirusFun< TSeq > fun)
- void **set_prob_death_fun** (VirusFun< TSeq > fun)
- void **set_incubation_fun** (VirusFun< TSeq > fun)
- void **set_prob_infecting** (const epiworld_double *prob)
- void **set_prob_recovery** (const epiworld_double *prob)
- void **set_prob_death** (const epiworld_double *prob)
- void **set_incubation** (const epiworld_double *prob)
- void **set_prob_infecting** (epiworld_double prob)
- void **set_prob_recovery** (epiworld_double prob)
- void **set_prob_death** (epiworld_double prob)
- void **set_incubation** (epiworld_double prob)

Get and set the state and queue

After applied, viruses can change the state and affect the queue of agents. These function sets the default values, which are retrieved when adding or removing a virus does not specify a change in state or in queue.

Parameters

init	After the virus/tool is added to the agent.
end	After the virus/tool is removed.
removed	After the agent (<i>Agent</i>) is removed.

- void **set_state** (epiworld_fast_int init, epiworld_fast_int end, epiworld_fast_int removed=-99)
- void **set_queue** (epiworld_fast_int init, epiworld_fast_int end, epiworld_fast_int removed=-99)
- void **get_state** (epiworld_fast_int *init, epiworld_fast_int *end, epiworld_fast_int *removed=nullptr)
- void **get_queue** (epiworld_fast_int *init, epiworld_fast_int *end, epiworld_fast_int *removed=nullptr)

- void **distribute** (*Model*< TSeq > *model)
Get information about the prevalence of the virus.
- void **set_distribution** (VirusToAgentFun< TSeq > fun)

Friends

- class **Agent**< TSeq >
- class **Model**< TSeq >
- class **DataBase**< TSeq >
- void **default_add_virus** (*Event*< TSeq > &a, *Model*< TSeq > *m)
- void **default_rm_virus** (*Event*< TSeq > &a, *Model*< TSeq > *m)

16.83.1 Detailed Description

```
template<typename TSeq>
class epiworld::Virus< TSeq >
```

Virus.

Template Parameters

<i>TSeq</i>	
-------------	--

Raw transmissibility of a virus should be a function of its genetic sequence. Nonetheless, transmissibility can be reduced as a result of having one or more tools to fight the virus. Because of this, transmissibility should be a function of the agent.

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

- epiworld.hpp

16.84 Virus< TSeq > Class Template Reference

[Virus.](#)

```
#include <virus-bones.hpp>
```

Public Member Functions

- **Virus** (std::string name="unknown virus")
- **Virus** (std::string name, epiworld_double prevalence, bool as_proportion)
- void **mutate** ([Model](#)< TSeq > *model)
- void **set_mutation** (MutFun< TSeq > fun)
- **EPI_TYPENAME_TRAITS** (TSeq, int) get_sequence()
- void **set_sequence** (TSeq sequence)
- [Agent](#)< TSeq > * **get_agent** ()
- void **set_agent** ([Agent](#)< TSeq > *p)
- void **set_date** (int d)
- int **get_date** () const
- void **set_id** (int idx)
- int **get_id** () const
- void **set_name** (std::string name)
- std::string **get_name** () const
- bool **operator==** (const [Virus](#)< TSeq > &other) const
- bool **operator!=** (const [Virus](#)< TSeq > &other) const
- void **print** () const
- void **set_sequence** (int sequence)
- bool **operator==** (const [Virus](#)< std::vector< int >> &other) const
- void **set_sequence** (int sequence)
- bool **operator==** (const [Virus](#)< std::vector< int >> &other) const

Get and set the tool functions

Parameters

v	<i>The virus over which to operate</i>
fun	<i>the function to be used</i>

Returns

epiworld_double

- epiworld_double **get_prob_infecting** ([Model](#)< TSeq > *model)
- epiworld_double **get_prob_recovery** ([Model](#)< TSeq > *model)
- epiworld_double **get_prob_death** ([Model](#)< TSeq > *model)
- epiworld_double **get_incubation** ([Model](#)< TSeq > *model)
- void **post_recovery** ([Model](#)< TSeq > *model)
- void **set_post_recovery** (PostRecoveryFun< TSeq > fun)
- void **set_post_immunity** (epiworld_double prob)
- void **set_prob_infecting** (epiworld_double *prob)
- void **set_prob_infecting_fun** (VirusFun< TSeq > fun)
- void **set_prob_recovery_fun** (VirusFun< TSeq > fun)
- void **set_prob_death_fun** (VirusFun< TSeq > fun)
- void **set_incubation_fun** (VirusFun< TSeq > fun)
- void **set_prob_infecting** (const epiworld_double *prob)
- void **set_prob_recovery** (const epiworld_double *prob)
- void **set_prob_death** (const epiworld_double *prob)
- void **set_incubation** (const epiworld_double *prob)
- void **set_prob_infecting** (epiworld_double prob)
- void **set_prob_recovery** (epiworld_double prob)
- void **set_prob_death** (epiworld_double prob)
- void **set_incubation** (epiworld_double prob)

Get and set the state and queue

After applied, viruses can change the state and affect the queue of agents. These function sets the default values, which are retrieved when adding or removing a virus does not specify a change in state or in queue.

Parameters

init	After the virus/tool is added to the agent.
end	After the virus/tool is removed.
removed	After the agent (Agent) is removed.

- void **set_state** (epiworld_fast_int init, epiworld_fast_int end, epiworld_fast_int removed=-99)
- void **set_queue** (epiworld_fast_int init, epiworld_fast_int end, epiworld_fast_int removed=-99)
- void **get_state** (epiworld_fast_int *init, epiworld_fast_int *end, epiworld_fast_int *removed=nullptr)
- void **get_queue** (epiworld_fast_int *init, epiworld_fast_int *end, epiworld_fast_int *removed=nullptr)

- void **distribute** ([Model](#)< TSeq > *model)
Get information about the prevalence of the virus.
- void **set_distribution** (VirusToAgentFun< TSeq > fun)

Friends

- class **Agent**< TSeq >
- class **Model**< TSeq >
- class **DataBase**< TSeq >
- void **default_add_virus** ([Event](#)< TSeq > &a, [Model](#)< TSeq > *m)
- void **default_rm_virus** ([Event](#)< TSeq > &a, [Model](#)< TSeq > *m)

16.84.1 Detailed Description

```
template<typename TSeq>
class Virus< TSeq >
```

[Virus.](#)

Template Parameters

<i>TSeq</i>	
-------------	--

Raw transmissibility of a virus should be a function of its genetic sequence. Nonetheless, transmissibility can be reduced as a result of having one or more tools to fight the virus. Because of this, transmissibility should be a function of the agent.

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

- include/epiworld/agent-bones.hpp
- include/epiworld/virus-bones.hpp
- include/epiworld/virus-meat.hpp

16.85 epiworld::Viruses< TSeq > Class Template Reference

Set of viruses (useful for building iterators)

```
#include <epiworld.hpp>
```

Public Member Functions

- **Viruses** ([Agent](#)< TSeq > &p)
- std::vector< VirusPtr< TSeq > >::iterator **begin** ()
- std::vector< VirusPtr< TSeq > >::iterator **end** ()
- VirusPtr< TSeq > & **operator**() (size_t i)
- VirusPtr< TSeq > & **operator[]** (size_t i)
- size_t **size** () const noexcept
- void **print** () const noexcept

Friends

- class **Virus**< TSeq >
- class **Agent**< TSeq >

16.85.1 Detailed Description

```
template<typename TSeq>
class epiworld::Viruses< TSeq >
```

Set of viruses (useful for building iterators)

Template Parameters

<i>TSeq</i>	
-------------	--

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

- epiworld.hpp

16.86 Viruses< TSeq > Class Template Reference

Set of viruses (useful for building iterators)

```
#include <viruses-bones.hpp>
```

Public Member Functions

- **Viruses** ([Agent](#)< TSeq > &p)
- std::vector< VirusPtr< TSeq > >::iterator **begin** ()
- std::vector< VirusPtr< TSeq > >::iterator **end** ()
- VirusPtr< TSeq > & **operator**() (size_t i)
- VirusPtr< TSeq > & **operator**[] (size_t i)
- size_t **size** () const noexcept
- void **print** () const noexcept

Friends

- class **Virus**< TSeq >
- class **Agent**< TSeq >

16.86.1 Detailed Description

```
template<typename TSeq>
class Viruses< TSeq >
```

Set of viruses (useful for building iterators)

Template Parameters

<i>TSeq</i>	
-------------	--

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

- include/epiworld/agent-bones.hpp
- include/epiworld/viruses-bones.hpp

16.87 epiworld::Viruses_const< TSeq > Class Template Reference

Set of [Viruses](#) (const) (useful for iterators)

```
#include <epiworld.hpp>
```

Public Member Functions

- **Viruses_const** (const [Agent](#)< TSeq > &p)
- std::vector< VirusPtr< TSeq > >::const_iterator **begin** () const
- std::vector< VirusPtr< TSeq > >::const_iterator **end** () const
- const VirusPtr< TSeq > & **operator()** (size_t i)
- const VirusPtr< TSeq > & **operator[]** (size_t i)
- size_t **size** () const noexcept
- void **print** () const noexcept

Friends

- class **Virus**< TSeq >
- class **Agent**< TSeq >

16.87.1 Detailed Description

```
template<typename TSeq>
class epiworld::Viruses_const< TSeq >
```

Set of [Viruses](#) (const) (useful for iterators)

Template Parameters

<i>TSeq</i>	
-------------	--

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

- epiworld.hpp

16.88 Viruses_const< TSeq > Class Template Reference

Set of [Viruses](#) (const) (useful for iterators)

```
#include <viruses-bones.hpp>
```

Public Member Functions

- **Viruses_const** (const [Agent](#)< TSeq > &p)
- std::vector< VirusPtr< TSeq > >::const_iterator **begin** () const
- std::vector< VirusPtr< TSeq > >::const_iterator **end** () const
- const VirusPtr< TSeq > & **operator**() (size_t i)
- const VirusPtr< TSeq > & **operator**[] (size_t i)
- size_t **size** () const noexcept
- void **print** () const noexcept

Friends

- class **Virus**< TSeq >
- class **Agent**< TSeq >

16.88.1 Detailed Description

```
template<typename TSeq>
class Viruses_const< TSeq >
```

Set of [Viruses](#) (const) (useful for iterators)

Template Parameters

<i>TSeq</i>	
-------------	--

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

- include/epiworld/agent-bones.hpp
- include/epiworld/viruses-bones.hpp

16.89 epiworld::VirusFunctions< TSeq > Class Template Reference

Public Attributes

- MutFun< TSeq > **mutation** = nullptr
- PostRecoveryFun< TSeq > **post_recovery** = nullptr
- VirusFun< TSeq > **probability_of_infecting** = nullptr
- VirusFun< TSeq > **probability_of_recovery** = nullptr
- VirusFun< TSeq > **probability_of_death** = nullptr
- VirusFun< TSeq > **incubation** = nullptr
- VirusToAgentFun< TSeq > **dist** = nullptr

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

- epiworld.hpp

16.90 VirusFunctions< TSeq > Class Template Reference

Public Attributes

- MutFun< TSeq > **mutation** = nullptr
- PostRecoveryFun< TSeq > **post_recovery** = nullptr
- VirusFun< TSeq > **probability_of_infecting** = nullptr
- VirusFun< TSeq > **probability_of_recovery** = nullptr
- VirusFun< TSeq > **probability_of_death** = nullptr
- VirusFun< TSeq > **incubation** = nullptr
- VirusToAgentFun< TSeq > **dist** = nullptr

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

- include/epiworld/virus-bones.hpp

Chapter 17

File Documentation

17.1 include/epiworld/agent-meat-state.hpp File Reference

Sampling functions are getting big, so we keep them in a separate file.

```
#include "model-bones.hpp"  
#include "agent-meat-virus-sampling.hpp"  
#include "config.hpp"
```

Include dependency graph for agent-meat-state.hpp:



This graph shows which files directly or indirectly include this file:



Functions

- `template<typename TSeq = EPI_DEFAULT_TSEQ>`
`void default_update_susceptible (Agent< TSeq > *p, Model< TSeq > *m)`
- `template<typename TSeq = EPI_DEFAULT_TSEQ>`
`void default_update_exposed (Agent< TSeq > *p, Model< TSeq > *m)`

17.1.1 Detailed Description

Sampling functions are getting big, so we keep them in a separate file.

Author

George G. Vega Yon (g.vegayon en gmail)

Version

0.1

Date

2022-06-15

Copyright

Copyright (c) 2022

17.2 include/epiworld/models/measlesmixing.hpp File Reference

Template for a Measles model with population mixing, quarantine, and contact tracing.

This graph shows which files directly or indirectly include this file:



Classes

- class [ModelMeaslesMixing< TSeq >](#)
Measles model with population mixing, quarantine, and contact tracing.

Macros

- `#define MM(i, j, n) j * n + i`
- `#define GET_MODEL(model, output)`
- `#define SAMPLE_FROM_PROBS(n, ans)`

17.2.1 Detailed Description

Template for a Measles model with population mixing, quarantine, and contact tracing.

17.2.2 Macro Definition Documentation

17.2.2.1 GET_MODEL

```
#define GET_MODEL(  
    model,  
    output )
```

Value:

```
auto * output = dynamic_cast< ModelMeaslesMixing<TSeq> * >( (model) ); \  
assert((output) != nullptr);
```

17.2.2.2 SAMPLE_FROM_PROBS

```
#define SAMPLE_FROM_PROBS(  
    n,  
    ans )
```

Value:

```
size_t ans; \  
epiworld_double p_total = m->runif(); \  
for (ans = 0u; ans < n; ++ans) \  
{ \  
    if (p_total < m->array_double_tmp[ans]) \  
        break; \  
    m->array_double_tmp[ans + 1] += m->array_double_tmp[ans]; \  
}
```

17.3 include/epiworld/models/seirmixingquarantine.hpp File Reference

Template for a Susceptible-Exposed-Infected-Removed (SEIR) model with mixing, quarantine, and contact tracing.

This graph shows which files directly or indirectly include this file:



Classes

- class [ModelSEIRMixingQuarantine< TSeq >](#)
SEIR model with mixing, quarantine, and contact tracing.

Macros

- `#define MM(i, j, n) j * n + i`
- `#define GET_MODEL(model, output)`
- `#define SAMPLE_FROM_PROBS(n, ans)`

17.3.1 Detailed Description

Template for a Susceptible-Exposed-Infected-Removed (SEIR) model with mixing, quarantine, and contact tracing.

17.3.2 Macro Definition Documentation

17.3.2.1 GET_MODEL

```
#define GET_MODEL(  
    model,  
    output )
```

Value:

```
auto * output = dynamic_cast< ModelSEIRMixingQuarantine<TSeq> * >( (model) ); \
assert((output) != nullptr);
```

17.3.2.2 SAMPLE_FROM_PROBS

```
#define SAMPLE_FROM_PROBS(  
    n,  
    ans )
```

Value:

```
size_t ans; \
epiworld_double p_total = m->runif(); \
for (ans = 0u; ans < n; ++ans) \
{ \
    if (p_total < m->array_double_tmp[ans]) \
        break; \
    m->array_double_tmp[ans + 1] += m->array_double_tmp[ans]; \
}
```


Index

add_globlevent
 epiworld::Model< TSeq >, 93
 Model< TSeq >, 109
AdjList, 45
 AdjList, 45
 epiworld::AdjList, 47
 read_edgelist, 46
Agent< TSeq >, 48
 default_rm_entity, 52
 operator(), 50
 swap_neighbors, 52
AgentsSample
 AgentsSample< TSeq >, 57
 epiworld::AgentsSample< TSeq >, 59
AgentsSample< TSeq >, 56
 AgentsSample, 57

clone_ptr
 epiworld::epimodels::ModelMeaslesMixing< TSeq
 >, 126
 epiworld::epimodels::ModelMeaslesSchool< TSeq
 >, 140
 epiworld::epimodels::ModelSEIRCONN< TSeq >,
 153
 epiworld::epimodels::ModelSEIRDCONN< TSeq
 >, 166
 epiworld::epimodels::ModelSEIRMixing< TSeq >,
 173
 epiworld::epimodels::ModelSEIRMixingQuarantine<
 TSeq >, 183
 epiworld::epimodels::ModelSIRCONN< TSeq >,
 199
 epiworld::epimodels::ModelSIRDCONN< TSeq >,
 212
 epiworld::epimodels::ModelSIRLogit< TSeq >,
 218
 epiworld::epimodels::ModelSIRMixing< TSeq >,
 224
 epiworld::Model< TSeq >, 93
 Model< TSeq >, 109
 ModelMeaslesMixing< TSeq >, 134
 ModelMeaslesSchool< TSeq >, 145
 ModelSEIRCONN< TSeq >, 156
 ModelSEIRDCONN< TSeq >, 169
 ModelSEIRMixing< TSeq >, 176
 ModelSEIRMixingQuarantine< TSeq >, 191
 ModelSIRCONN< TSeq >, 203
 ModelSIRDCONN< TSeq >, 214
 ModelSIRLogit< TSeq >, 221
 ModelSIRMixing< TSeq >, 228

DataBase< TSeq >, 59
 get_generation_time, 62
 get_reproductive_number, 62
 get_transition_probability, 62
 get_transmissions, 63
 operator==, 63, 64
 record_virus, 64
default_rm_entity
 Agent< TSeq >, 52
 Entity< TSeq >, 74
 epiworld::Agent< TSeq >, 56
 epiworld::Entity< TSeq >, 75
draw
 epiworld::Model< TSeq >, 94
 Model< TSeq >, 109

Entities< TSeq >, 69
Entities_const< TSeq >, 71
Entity
 Entity< TSeq >, 73
 epiworld::Entity< TSeq >, 75
Entity< TSeq >, 73
 default_rm_entity, 74
 Entity, 73
epiworld::AdjList, 46
 AdjList, 47
 read_edgelist, 47
epiworld::Agent< TSeq >, 53
 default_rm_entity, 56
 operator(), 55
 swap_neighbors, 55
epiworld::AgentsSample< TSeq >, 58
 AgentsSample, 59
epiworld::DataBase< TSeq >, 65
 get_generation_time, 67
 get_reproductive_number, 67
 get_transition_probability, 68
 get_transmissions, 68
 operator==, 69
 record_virus, 69
epiworld::Entities< TSeq >, 70
epiworld::Entities_const< TSeq >, 72
epiworld::Entity< TSeq >, 74
 default_rm_entity, 75
 Entity, 75
epiworld::epimodels::ModelDiffNet< TSeq >, 117
epiworld::epimodels::ModelMeaslesMixing< TSeq >,
 120
 clone_ptr, 126
 get_agent_quarantine_triggered, 126

- get_contact_matrix, 126
 - get_isolation_willingness, 127
 - get_quarantine_willingness, 127
 - initial_states, 127
 - ModelMeaslesMixing, 123, 125
 - run, 128
 - set_contact_matrix, 128
- epiworld::epimodels::ModelMeaslesSchool< TSeq >, 137
 - clone_ptr, 140
 - ModelMeaslesSchool, 139
 - quarantine_agents, 141
 - reset, 141
- epiworld::epimodels::ModelSEIR< TSeq >, 146
 - initial_states, 148
 - update_exposed_seir, 148
 - update_infected_seir, 148
- epiworld::epimodels::ModelSEIRCONN< TSeq >, 152
 - clone_ptr, 153
 - initial_states, 154
 - ModelSEIRCONN, 153
 - reset, 154
- epiworld::epimodels::ModelSEIRD< TSeq >, 158
 - ModelSEIRD, 159, 160
 - update_exposed_seir, 160
- epiworld::epimodels::ModelSEIRDCONN< TSeq >, 164
 - clone_ptr, 166
 - initial_states, 166
 - ModelSEIRDCONN, 165
 - reset, 166
- epiworld::epimodels::ModelSEIRMixing< TSeq >, 170
 - clone_ptr, 173
 - initial_states, 173
 - ModelSEIRMixing, 171, 172
 - reset, 173
- epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >, 178
 - clone_ptr, 183
 - get_agent_quarantine_triggered, 183
 - get_contact_matrix, 183
 - get_isolation_willingness, 184
 - get_quarantine_willingness, 184
 - initial_states, 184
 - ModelSEIRMixingQuarantine, 180, 182
 - run, 185
 - set_contact_matrix, 185
- epiworld::epimodels::ModelSIR< TSeq >, 194
 - initial_states, 195
- epiworld::epimodels::ModelSIRCONN< TSeq >, 198
 - clone_ptr, 199
 - get_n_infected, 200
 - initial_states, 200
 - ModelSIRCONN, 199
 - reset, 200
- epiworld::epimodels::ModelSIRD< TSeq >, 204
 - initial_states, 207
 - ModelSIRD, 205
- epiworld::epimodels::ModelSIRDCONN< TSeq >, 210
 - clone_ptr, 212
 - ModelSIRDCONN, 211
 - reset, 212
- epiworld::epimodels::ModelSIRLogit< TSeq >, 215
 - clone_ptr, 218
 - ModelSIRLogit, 217
 - reset, 218
- epiworld::epimodels::ModelSIRMixing< TSeq >, 222
 - clone_ptr, 224
 - initial_states, 225
 - ModelSIRMixing, 223, 224
 - reset, 225
- epiworld::epimodels::ModelSIS< TSeq >, 229
- epiworld::epimodels::ModelSISD< TSeq >, 233
- epiworld::epimodels::ModelSURV< TSeq >, 236
 - reset, 238
- epiworld::Event< TSeq >, 76
 - Event, 77
- epiworld::GlobalEvent< TSeq >, 79
 - GlobalEvent, 80
- epiworld::LFMCMC< TData >, 82
- epiworld::Model< TSeq >, 85
 - add_globalevent, 93
 - clone_ptr, 93
 - draw, 94
 - events_add, 94
 - events_run, 95
 - initial_states_fun, 98
 - load_agents_entities_ties, 95
 - rbinomd, 98
 - reset, 95
 - rexp, 98
 - rgammad, 98
 - rgeomd, 98
 - rlognormald, 99
 - rnbinomd, 99
 - rnormd, 99
 - rpoissd, 99
 - run_multiple, 96
 - runifd, 99
 - set_agents_data, 96
 - set_name, 97
 - time_elapsed, 100
 - write_data, 97
- epiworld::ModelDiagram, 117
- epiworld::PersonTools< TSeq >, 241
- epiworld::Progress, 242
- epiworld::Queue< TSeq >, 243
- epiworld::sampler, 37
 - make_sample_virus_neighbors, 37
 - make_update_susceptible, 38
 - sample_virus_single, 38
- epiworld::SAMPLETYPE, 245
- epiworld::Tool< TSeq >, 245
- epiworld::ToolFunctions< TSeq >, 248
- epiworld::Tools< TSeq >, 249
- epiworld::Tools_const< TSeq >, 251

- epiworld::UserData< TSeq >, 252
 - UserData, 254
- epiworld::vecHasher< T >, 256
- epiworld::Virus< TSeq >, 257
- epiworld::Viruses< TSeq >, 261
- epiworld::Viruses_const< TSeq >, 263
- epiworld::VirusFunctions< TSeq >, 264
- Event
 - epiworld::Event< TSeq >, 77
 - Event< TSeq >, 79
- Event< TSeq >, 77
 - Event, 79
- events_add
 - epiworld::Model< TSeq >, 94
 - Model< TSeq >, 110
- events_run
 - epiworld::Model< TSeq >, 95
 - Model< TSeq >, 110
- get_agent_quarantine_triggered
 - epiworld::epimodels::ModelMeaslesMixing< TSeq >, 126
 - epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >, 183
 - ModelMeaslesMixing< TSeq >, 135
 - ModelSEIRMixingQuarantine< TSeq >, 191
- get_contact_matrix
 - epiworld::epimodels::ModelMeaslesMixing< TSeq >, 126
 - epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >, 183
 - ModelMeaslesMixing< TSeq >, 135
 - ModelSEIRMixingQuarantine< TSeq >, 191
- get_generation_time
 - DataBase< TSeq >, 62
 - epiworld::DataBase< TSeq >, 67
- get_isolation_willingness
 - epiworld::epimodels::ModelMeaslesMixing< TSeq >, 127
 - epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >, 184
 - ModelMeaslesMixing< TSeq >, 135
 - ModelSEIRMixingQuarantine< TSeq >, 192
- GET_MODEL
 - measlesmixing.hpp, 269
 - seirmixingquarantine.hpp, 271
- get_n_infected
 - epiworld::epimodels::ModelSIRCONN< TSeq >, 200
 - ModelSIRCONN< TSeq >, 203
- get_quarantine_willingness
 - epiworld::epimodels::ModelMeaslesMixing< TSeq >, 127
 - epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >, 184
 - ModelMeaslesMixing< TSeq >, 135
 - ModelSEIRMixingQuarantine< TSeq >, 192
- get_reproductive_number
 - DataBase< TSeq >, 62
- epiworld::DataBase< TSeq >, 67
- get_transition_probability
 - DataBase< TSeq >, 62
 - epiworld::DataBase< TSeq >, 68
- get_transmissions
 - DataBase< TSeq >, 63
 - epiworld::DataBase< TSeq >, 68
- GlobalEvent
 - epiworld::GlobalEvent< TSeq >, 80
 - GlobalEvent< TSeq >, 81
- GlobalEvent< TSeq >, 81
 - GlobalEvent, 81
- include/epiworld/agent-meat-state.hpp, 267
- include/epiworld/models/measlesmixing.hpp, 269
- include/epiworld/models/seirmixingquarantine.hpp, 270
- initial_states
 - epiworld::epimodels::ModelMeaslesMixing< TSeq >, 127
 - epiworld::epimodels::ModelSEIR< TSeq >, 148
 - epiworld::epimodels::ModelSEIRCONN< TSeq >, 154
 - epiworld::epimodels::ModelSEIRDCONN< TSeq >, 166
 - epiworld::epimodels::ModelSEIRMixing< TSeq >, 173
 - epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >, 184
 - epiworld::epimodels::ModelSIR< TSeq >, 195
 - epiworld::epimodels::ModelSIRCONN< TSeq >, 200
 - epiworld::epimodels::ModelSIRD< TSeq >, 207
 - epiworld::epimodels::ModelSIRMixing< TSeq >, 225
 - ModelMeaslesMixing< TSeq >, 136
 - ModelSEIR< TSeq >, 150
 - ModelSEIRCONN< TSeq >, 157
 - ModelSEIRDCONN< TSeq >, 169
 - ModelSEIRMixing< TSeq >, 177
 - ModelSEIRMixingQuarantine< TSeq >, 192
 - ModelSIR< TSeq >, 197
 - ModelSIRCONN< TSeq >, 203
 - ModelSIRD< TSeq >, 209
 - ModelSIRMixing< TSeq >, 228
- initial_states_fun
 - epiworld::Model< TSeq >, 98
 - Model< TSeq >, 114
- LFMCMC< TData >, 83
- load_agents_entities_ties
 - epiworld::Model< TSeq >, 95
 - Model< TSeq >, 112
- make_sample_virus_neighbors
 - epiworld::sampler, 37
 - sampler, 40
- make_update_susceptible
 - epiworld::sampler, 38
 - sampler, 41

- measlesmixing.hpp
 - GET_MODEL, 269
 - SAMPLE_FROM_PROBS, 270
- Model< TSeq >, 100
 - add_globlevent, 109
 - clone_ptr, 109
 - draw, 109
 - events_add, 110
 - events_run, 110
 - initial_states_fun, 114
 - load_agents_entities_ties, 112
 - rbinomd, 114
 - reset, 112
 - rexp, 114
 - rgammad, 115
 - rgeomd, 115
 - rlognormald, 115
 - rnbinomd, 115
 - rnormd, 115
 - rpoissd, 116
 - run_multiple, 112
 - runifd, 116
 - set_agents_data, 113
 - set_name, 113
 - time_elapsed, 116
 - write_data, 114
- ModelDiagram, 117
- ModelDiffNet< TSeq >, 119
- ModelMeaslesMixing
 - epiworld::epimodels::ModelMeaslesMixing< TSeq >, 123, 125
 - ModelMeaslesMixing< TSeq >, 132, 133
- ModelMeaslesMixing< TSeq >, 129
 - clone_ptr, 134
 - get_agent_quarantine_triggered, 135
 - get_contact_matrix, 135
 - get_isolation_willingness, 135
 - get_quarantine_willingness, 135
 - initial_states, 136
 - ModelMeaslesMixing, 132, 133
 - run, 136
 - set_contact_matrix, 137
- ModelMeaslesSchool
 - epiworld::epimodels::ModelMeaslesSchool< TSeq >, 139
 - ModelMeaslesSchool< TSeq >, 144
- ModelMeaslesSchool< TSeq >, 142
 - clone_ptr, 145
 - ModelMeaslesSchool, 144
 - quarantine_agents, 145
 - reset, 145
- ModelSEIR< TSeq >, 149
 - initial_states, 150
 - update_exposed_seir, 151
 - update_infected_seir, 151
- ModelSEIRCONN
 - epiworld::epimodels::ModelSEIRCONN< TSeq >, 153
 - ModelSEIRCONN< TSeq >, 156
 - clone_ptr, 156
 - initial_states, 157
 - ModelSEIRCONN, 156
 - reset, 157
- ModelSEIRD
 - epiworld::epimodels::ModelSEIRD< TSeq >, 159, 160
 - ModelSEIRD< TSeq >, 162, 163
- ModelSEIRD< TSeq >, 161
 - ModelSEIRD, 162, 163
 - update_exposed_seir, 163
- ModelSEIRDCONN
 - epiworld::epimodels::ModelSEIRDCONN< TSeq >, 165
 - ModelSEIRDCONN< TSeq >, 168
- ModelSEIRDCONN< TSeq >, 167
 - clone_ptr, 169
 - initial_states, 169
 - ModelSEIRDCONN, 168
 - reset, 170
- ModelSEIRMixing
 - epiworld::epimodels::ModelSEIRMixing< TSeq >, 171, 172
 - ModelSEIRMixing< TSeq >, 175, 176
- ModelSEIRMixing< TSeq >, 174
 - clone_ptr, 176
 - initial_states, 177
 - ModelSEIRMixing, 175, 176
 - reset, 177
- ModelSEIRMixingQuarantine
 - epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >, 180, 182
 - ModelSEIRMixingQuarantine< TSeq >, 188, 190
- ModelSEIRMixingQuarantine< TSeq >, 186
 - clone_ptr, 191
 - get_agent_quarantine_triggered, 191
 - get_contact_matrix, 191
 - get_isolation_willingness, 192
 - get_quarantine_willingness, 192
 - initial_states, 192
 - ModelSEIRMixingQuarantine, 188, 190
 - run, 193
 - set_contact_matrix, 193
- ModelSIR< TSeq >, 195
 - initial_states, 197
- ModelSIRCONN
 - epiworld::epimodels::ModelSIRCONN< TSeq >, 199
 - ModelSIRCONN< TSeq >, 202
- ModelSIRCONN< TSeq >, 201
 - clone_ptr, 203
 - get_n_infected, 203
 - initial_states, 203
 - ModelSIRCONN, 202
 - reset, 204
- ModelSIRD

- epiworld::epimodels::ModelSIRD< TSeq >, 205
- ModelSIRD< TSeq >, 209
- ModelSIRD< TSeq >, 207
 - initial_states, 209
 - ModelSIRD, 209
- ModelSIRDCONN
 - epiworld::epimodels::ModelSIRDCONN< TSeq >, 211
 - ModelSIRDCONN< TSeq >, 214
- ModelSIRDCONN< TSeq >, 213
 - clone_ptr, 214
 - ModelSIRDCONN, 214
 - reset, 215
- ModelSIRLogit
 - epiworld::epimodels::ModelSIRLogit< TSeq >, 217
 - ModelSIRLogit< TSeq >, 220
- ModelSIRLogit< TSeq >, 219
 - clone_ptr, 221
 - ModelSIRLogit, 220
 - reset, 221
- ModelSIRMixing
 - epiworld::epimodels::ModelSIRMixing< TSeq >, 223, 224
 - ModelSIRMixing< TSeq >, 227, 228
- ModelSIRMixing< TSeq >, 226
 - clone_ptr, 228
 - initial_states, 228
 - ModelSIRMixing, 227, 228
 - reset, 229
- ModelSIS< TSeq >, 232
- ModelSISD< TSeq >, 235
- ModelSURV< TSeq >, 239
 - reset, 240
- Network< Nettype, Nodetype, Edgetype >, 241
- operator()
 - Agent< TSeq >, 50
 - epiworld::Agent< TSeq >, 55
- operator==
 - DataBase< TSeq >, 63, 64
 - epiworld::DataBase< TSeq >, 69
- PersonTools< TSeq >, 242
- Progress, 242
- quarantine_agents
 - epiworld::epimodels::ModelMeaslesSchool< TSeq >, 141
 - ModelMeaslesSchool< TSeq >, 145
- Queue< TSeq >, 243
- RandGraph, 244
- rbinomd
 - epiworld::Model< TSeq >, 98
 - Model< TSeq >, 114
- read_edgelist
 - AdjList, 46
- epiworld::AdjList, 47
- record_virus
 - DataBase< TSeq >, 64
 - epiworld::DataBase< TSeq >, 69
- reset
 - epiworld::epimodels::ModelMeaslesSchool< TSeq >, 141
 - epiworld::epimodels::ModelSEIRCONN< TSeq >, 154
 - epiworld::epimodels::ModelSEIRDCONN< TSeq >, 166
 - epiworld::epimodels::ModelSEIRMixing< TSeq >, 173
 - epiworld::epimodels::ModelSIRCONN< TSeq >, 200
 - epiworld::epimodels::ModelSIRDCONN< TSeq >, 212
 - epiworld::epimodels::ModelSIRLogit< TSeq >, 218
 - epiworld::epimodels::ModelSIRMixing< TSeq >, 225
 - epiworld::epimodels::ModelSURV< TSeq >, 238
 - epiworld::Model< TSeq >, 95
 - Model< TSeq >, 112
 - ModelMeaslesSchool< TSeq >, 145
 - ModelSEIRCONN< TSeq >, 157
 - ModelSEIRDCONN< TSeq >, 170
 - ModelSEIRMixing< TSeq >, 177
 - ModelSIRCONN< TSeq >, 204
 - ModelSIRDCONN< TSeq >, 215
 - ModelSIRLogit< TSeq >, 221
 - ModelSIRMixing< TSeq >, 229
 - ModelSURV< TSeq >, 240
- rexp
 - epiworld::Model< TSeq >, 98
 - Model< TSeq >, 114
- rgammad
 - epiworld::Model< TSeq >, 98
 - Model< TSeq >, 115
- rgeomd
 - epiworld::Model< TSeq >, 98
 - Model< TSeq >, 115
- rlognormald
 - epiworld::Model< TSeq >, 99
 - Model< TSeq >, 115
- rnbinomd
 - epiworld::Model< TSeq >, 99
 - Model< TSeq >, 115
- rnrm
 - epiworld::Model< TSeq >, 99
 - Model< TSeq >, 115
- rpoissd
 - epiworld::Model< TSeq >, 99
 - Model< TSeq >, 116
- run
 - epiworld::epimodels::ModelMeaslesMixing< TSeq >, 128

- epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >, 185
- ModelMeaslesMixing< TSeq >, 136
- ModelSEIRMixingQuarantine< TSeq >, 193
- run_multiple
 - epiworld::Model< TSeq >, 96
 - Model< TSeq >, 112
- runifd
 - epiworld::Model< TSeq >, 99
 - Model< TSeq >, 116
- SAMPLE_FROM_PROBS
 - measlesmixing.hpp, 270
 - seirmixingquarantine.hpp, 271
- sample_virus_single
 - epiworld::sampler, 38
 - sampler, 41
- sampler, 40
 - make_sample_virus_neighbors, 40
 - make_update_susceptible, 41
 - sample_virus_single, 41
- SAMPLETYPE, 245
- seirmixingquarantine.hpp
 - GET_MODEL, 271
 - SAMPLE_FROM_PROBS, 271
- set_agents_data
 - epiworld::Model< TSeq >, 96
 - Model< TSeq >, 113
- set_contact_matrix
 - epiworld::epimodels::ModelMeaslesMixing< TSeq >, 128
 - epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >, 185
 - ModelMeaslesMixing< TSeq >, 137
 - ModelSEIRMixingQuarantine< TSeq >, 193
- set_name
 - epiworld::Model< TSeq >, 97
 - Model< TSeq >, 113
- swap_neighbors
 - Agent< TSeq >, 52
 - epiworld::Agent< TSeq >, 55
- time_elapsed
 - epiworld::Model< TSeq >, 100
 - Model< TSeq >, 116
- Tool< TSeq >, 246
- ToolFunctions< TSeq >, 249
- Tools< TSeq >, 250
- Tools_const< TSeq >, 251
- update_exposed_seir
 - epiworld::epimodels::ModelSEIR< TSeq >, 148
 - epiworld::epimodels::ModelSEIRD< TSeq >, 160
 - ModelSEIR< TSeq >, 151
 - ModelSEIRD< TSeq >, 163
- update_infected_seir
 - epiworld::epimodels::ModelSEIR< TSeq >, 148
 - ModelSEIR< TSeq >, 151
- UserData
 - epiworld::UserData< TSeq >, 254
 - UserData< TSeq >, 255
 - UserData< TSeq >, 254
 - UserData, 255
- vecHasher< T >, 256
- Virus< TSeq >, 259
- Viruses< TSeq >, 262
- Viruses_const< TSeq >, 263
- VirusFunctions< TSeq >, 265
- write_data
 - epiworld::Model< TSeq >, 97
 - Model< TSeq >, 114