

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 ModelMeaslesMixingRiskQuarantine	11
5.1 Overview	11
5.1.1 Key Features	11
5.2 Disease States	11
5.3 Risk Classification System	12
5.3.1 High Risk	12
5.3.2 Medium Risk	12
5.3.3 Low Risk	12
5.4 Enhanced Detection	13
5.5 Parameters	13
5.5.1 Disease Parameters	13
5.5.2 Detection and Isolation	13
5.5.3 Risk-based Quarantine	13
5.5.4 Contact Tracing	14
5.5.5 Vaccination	14
5.6 Usage Example	14
5.7 Differences from Other Models	14
5.7.1 vs. ModelMeaslesMixing	14
5.7.2 vs. ModelMeaslesSchool	15
5.8 Testing	15

5.9 Applications	15
5.10 Implementation Notes	15
6 epiworld c++ template library	17
6.1 Main features	17
6.2 Algorithm	17
6.3 Hello world (C++)	18
6.4 Surveillance simulation	18
6.4.1 Preliminary results	19
6.4.2 Cases detected	20
7 General parameters	21
7.1 Compartmental Models	21
7.1.1 SIR Model	21
7.1.2 SEIR Model	22
7.2 Agent-Based Model Approach	22
7.2.1 Mathematical preliminaries	23
7.2.2 Simulation study	23
7.3 Comparing ABM with Compartmental Models	23
7.3.1 SIR	23
7.3.2 SEIR	24
7.3.3 Rates	24
8 MIT License	25
9 model1	27
10 Mixing probabilities in connected model	29
10.1 Case 1: No grouping	29
10.2 Case 2: Grouping	30
11 EPI Simulator	31
11.1 Disease dynamics	31
11.2 Network dynamics	31
11.3 Contagion dynamics	31
11.4 Time dynamics	31
11.5 Updating agent's status	32
11.5.1 Other parameters	32
12 Namespace Index	33
12.1 Namespace List	33
13 Hierarchical Index	35
13.1 Class Hierarchy	35
14 Class Index	37

14.1 Class List	37
15 File Index	41
15.1 File List	41
16 Namespace Documentation	43
16.1 epiworld::sampler Namespace Reference	43
16.1.1 Detailed Description	43
16.1.2 Function Documentation	43
16.1.2.1 make_sample_virus_neighbors()	43
16.1.2.2 make_update_susceptible()	44
16.1.2.3 sample_virus_single()	44
16.2 sampler Namespace Reference	46
16.2.1 Detailed Description	46
16.2.2 Function Documentation	46
16.2.2.1 make_sample_virus_neighbors()	46
16.2.2.2 make_update_susceptible()	47
16.2.2.3 sample_virus_single()	47
17 Class Documentation	51
17.1 AdjList Class Reference	51
17.1.1 Constructor & Destructor Documentation	51
17.1.1.1 AdjList()	51
17.1.2 Member Function Documentation	52
17.1.2.1 read_edgelist()	52
17.2 epiworld::AdjList Class Reference	52
17.2.1 Constructor & Destructor Documentation	53
17.2.1.1 AdjList()	53
17.2.2 Member Function Documentation	53
17.2.2.1 read_edgelist()	53
17.3 Agent< TSeq > Class Template Reference	54
17.3.1 Detailed Description	56
17.3.2 Member Function Documentation	56
17.3.2.1 operator>()	56
17.3.2.2 swap_neighbors()	58
17.3.3 Friends And Related Function Documentation	58
17.3.3.1 default_rm_entity	58
17.4 epiworld::Agent< TSeq > Class Template Reference	59
17.4.1 Detailed Description	61
17.4.2 Member Function Documentation	61
17.4.2.1 operator>()	61
17.4.2.2 swap_neighbors()	62
17.4.3 Friends And Related Function Documentation	62

17.4.3.1 default_rm_entity	62
17.5 AgentsSample< TSeq > Class Template Reference	62
17.5.1 Detailed Description	63
17.5.2 Constructor & Destructor Documentation	63
17.5.2.1 AgentsSample()	63
17.6 epiworld::AgentsSample< TSeq > Class Template Reference	64
17.6.1 Detailed Description	64
17.6.2 Constructor & Destructor Documentation	65
17.6.2.1 AgentsSample()	65
17.7 DataBase< TSeq > Class Template Reference	65
17.7.1 Detailed Description	67
17.7.2 Member Function Documentation	68
17.7.2.1 get_generation_time()	68
17.7.2.2 get_reproductive_number()	68
17.7.2.3 get_transition_probability()	69
17.7.2.4 get_transmissions()	69
17.7.2.5 operator==() [1/3]	69
17.7.2.6 operator==() [2/3]	70
17.7.2.7 operator==() [3/3]	70
17.7.2.8 record_virus()	70
17.8 epiworld::DataBase< TSeq > Class Template Reference	71
17.8.1 Detailed Description	73
17.8.2 Member Function Documentation	73
17.8.2.1 get_generation_time()	73
17.8.2.2 get_reproductive_number()	73
17.8.2.3 get_transition_probability()	74
17.8.2.4 get_transmissions()	74
17.8.2.5 operator==()	75
17.8.2.6 record_virus()	75
17.9 Entities< TSeq > Class Template Reference	75
17.9.1 Detailed Description	76
17.10 epiworld::Entities< TSeq > Class Template Reference	76
17.10.1 Detailed Description	77
17.11 Entities_const< TSeq > Class Template Reference	77
17.11.1 Detailed Description	77
17.12 epiworld::Entities_const< TSeq > Class Template Reference	78
17.12.1 Detailed Description	78
17.13 Entity< TSeq > Class Template Reference	79
17.13.1 Constructor & Destructor Documentation	79
17.13.1.1 Entity()	79
17.13.2 Friends And Related Function Documentation	80
17.13.2.1 default_rm_entity	80

17.14 <code>epiworld::Entity< TSeq ></code> Class Template Reference	80
17.14.1 Constructor & Destructor Documentation	81
17.14.1.1 <code>Entity()</code>	81
17.14.2 Friends And Related Function Documentation	81
17.14.2.1 <code>default_rm_entity</code>	82
17.15 <code>epiworld::Event< TSeq ></code> Struct Template Reference	82
17.15.1 Detailed Description	82
17.15.2 Constructor & Destructor Documentation	83
17.15.2.1 <code>Event()</code>	83
17.16 <code>Event< TSeq ></code> Struct Template Reference	83
17.16.1 Detailed Description	84
17.16.2 Constructor & Destructor Documentation	85
17.16.2.1 <code>Event()</code>	85
17.17 <code>epiworld::GlobalEvent< TSeq ></code> Class Template Reference	85
17.17.1 Detailed Description	86
17.17.2 Constructor & Destructor Documentation	86
17.17.2.1 <code>GlobalEvent()</code>	86
17.18 <code>GlobalEvent< TSeq ></code> Class Template Reference	87
17.18.1 Detailed Description	87
17.18.2 Constructor & Destructor Documentation	87
17.18.2.1 <code>GlobalEvent()</code>	87
17.19 <code>epiworld::LFMCMC< TData ></code> Class Template Reference	88
17.19.1 Detailed Description	89
17.20 <code>LFMCMC< TData ></code> Class Template Reference	89
17.20.1 Detailed Description	90
17.21 <code>epiworld::Model< TSeq ></code> Class Template Reference	91
17.21.1 Detailed Description	99
17.21.2 Member Function Documentation	99
17.21.2.1 <code>add_globalevent()</code>	99
17.21.2.2 <code>clone_ptr()</code>	100
17.21.2.3 <code>draw()</code>	100
17.21.2.4 <code>events_add()</code>	100
17.21.2.5 <code>events_run()</code>	101
17.21.2.6 <code>load_agents_entities_ties()</code>	101
17.21.2.7 <code>reset()</code>	102
17.21.2.8 <code>run_multiple()</code>	102
17.21.2.9 <code>set_agents_data()</code>	102
17.21.2.10 <code>set_name()</code>	103
17.21.2.11 <code>write_data()</code>	103
17.21.3 Member Data Documentation	104
17.21.3.1 <code>initial_states_fun</code>	104
17.21.3.2 <code>rbinomd</code>	104

17.21.3.3 rexp	104
17.21.3.4 rgammad	104
17.21.3.5 rgeomd	105
17.21.3.6 rlognormald	105
17.21.3.7 rnbinomd	105
17.21.3.8 rnormd	105
17.21.3.9 rpoissd	105
17.21.3.10 runifd	106
17.21.3.11 time_elapsed	106
17.22 Model< TSeq > Class Template Reference	106
17.22.1 Detailed Description	114
17.22.2 Member Function Documentation	115
17.22.2.1 add_globlevent()	115
17.22.2.2 clone_ptr()	115
17.22.2.3 draw()	115
17.22.2.4 events_add()	116
17.22.2.5 events_run()	116
17.22.2.6 load_agents_entities_ties()	118
17.22.2.7 reset()	118
17.22.2.8 run_multiple()	119
17.22.2.9 set_agents_data()	119
17.22.2.10 set_name()	119
17.22.2.11 write_data()	120
17.22.3 Member Data Documentation	120
17.22.3.1 initial_states_fun	120
17.22.3.2 rbinomd	120
17.22.3.3 rexp	121
17.22.3.4 rgammad	121
17.22.3.5 rgeomd	121
17.22.3.6 rlognormald	121
17.22.3.7 rnbinomd	121
17.22.3.8 rnormd	122
17.22.3.9 rpoissd	122
17.22.3.10 runifd	122
17.22.3.11 time_elapsed	122
17.23 epiworld::ModelDiagram Class Reference	123
17.24 ModelDiagram Class Reference	123
17.25 epiworld::epimodels::ModelDiffNet< TSeq > Class Template Reference	123
17.25.1 Detailed Description	124
17.26 ModelDiffNet< TSeq > Class Template Reference	125
17.26.1 Detailed Description	126
17.27 epiworld::epimodels::ModelMeaslesMixing< TSeq > Class Template Reference	126

17.27.1 Detailed Description	128
17.27.2 Constructor & Destructor Documentation	129
17.27.2.1 ModelMeaslesMixing() [1/2]	129
17.27.2.2 ModelMeaslesMixing() [2/2]	131
17.27.3 Member Function Documentation	132
17.27.3.1 clone_ptr()	132
17.27.3.2 get_agent_quarantine_triggered()	132
17.27.3.3 get_contact_matrix()	133
17.27.3.4 get_isolation_willingness()	133
17.27.3.5 get_quarantine_willingness()	133
17.27.3.6 initial_states()	133
17.27.3.7 run()	134
17.27.3.8 set_contact_matrix()	134
17.28 ModelMeaslesMixing< TSeq > Class Template Reference	135
17.28.1 Detailed Description	137
17.28.2 Constructor & Destructor Documentation	138
17.28.2.1 ModelMeaslesMixing() [1/2]	138
17.28.2.2 ModelMeaslesMixing() [2/2]	139
17.28.3 Member Function Documentation	140
17.28.3.1 clone_ptr()	140
17.28.3.2 get_agent_quarantine_triggered()	141
17.28.3.3 get_contact_matrix()	141
17.28.3.4 get_isolation_willingness()	141
17.28.3.5 get_quarantine_willingness()	142
17.28.3.6 initial_states()	142
17.28.3.7 run()	142
17.28.3.8 set_contact_matrix()	143
17.29 epiworld::epimodels::ModelMeaslesSchool< TSeq > Class Template Reference	143
17.29.1 Detailed Description	145
17.29.2 Constructor & Destructor Documentation	145
17.29.2.1 ModelMeaslesSchool()	145
17.29.3 Member Function Documentation	146
17.29.3.1 clone_ptr()	146
17.29.3.2 quarantine_agents()	147
17.29.3.3 reset()	147
17.30 ModelMeaslesSchool< TSeq > Class Template Reference	148
17.30.1 Detailed Description	149
17.30.2 Constructor & Destructor Documentation	150
17.30.2.1 ModelMeaslesSchool()	150
17.30.3 Member Function Documentation	151
17.30.3.1 clone_ptr()	151
17.30.3.2 quarantine_agents()	151

17.30.3.3 reset()	152
17.31 epiworld::epimodels::ModelSEIR< TSeq > Class Template Reference	152
17.31.1 Detailed Description	153
17.31.2 Member Function Documentation	154
17.31.2.1 initial_states()	154
17.31.3 Member Data Documentation	154
17.31.3.1 update_exposed_seir	154
17.31.3.2 update_infected_seir	155
17.32 ModelSEIR< TSeq > Class Template Reference	155
17.32.1 Detailed Description	156
17.32.2 Member Function Documentation	156
17.32.2.1 initial_states()	157
17.32.3 Member Data Documentation	157
17.32.3.1 update_exposed_seir	157
17.32.3.2 update_infected_seir	157
17.33 epiworld::epimodels::ModelSEIRCONN< TSeq > Class Template Reference	158
17.33.1 Constructor & Destructor Documentation	159
17.33.1.1 ModelSEIRCONN()	159
17.33.2 Member Function Documentation	159
17.33.2.1 clone_ptr()	160
17.33.2.2 initial_states()	160
17.33.2.3 reset()	160
17.34 ModelSEIRCONN< TSeq > Class Template Reference	161
17.34.1 Constructor & Destructor Documentation	162
17.34.1.1 ModelSEIRCONN()	162
17.34.2 Member Function Documentation	162
17.34.2.1 clone_ptr()	163
17.34.2.2 initial_states()	163
17.34.2.3 reset()	163
17.35 epiworld::epimodels::ModelSEIRD< TSeq > Class Template Reference	164
17.35.1 Detailed Description	165
17.35.2 Constructor & Destructor Documentation	165
17.35.2.1 ModelSEIRD() ^[1/2]	165
17.35.2.2 ModelSEIRD() ^[2/2]	166
17.35.3 Member Data Documentation	166
17.35.3.1 update_exposed_seir	166
17.36 ModelSEIRD< TSeq > Class Template Reference	167
17.36.1 Detailed Description	168
17.36.2 Constructor & Destructor Documentation	168
17.36.2.1 ModelSEIRD() ^[1/2]	168
17.36.2.2 ModelSEIRD() ^[2/2]	169
17.36.3 Member Data Documentation	169

17.36.3.1 update_exposed_seir	169
17.37 epiworld::epimodels::ModelSEIRDCONN< TSeq > Class Template Reference	170
17.37.1 Constructor & Destructor Documentation	171
17.37.1.1 ModelSEIRDCONN()	171
17.37.2 Member Function Documentation	172
17.37.2.1 clone_ptr()	172
17.37.2.2 initial_states()	172
17.37.2.3 reset()	172
17.38 ModelSEIRDCONN< TSeq > Class Template Reference	173
17.38.1 Constructor & Destructor Documentation	174
17.38.1.1 ModelSEIRDCONN()	174
17.38.2 Member Function Documentation	175
17.38.2.1 clone_ptr()	175
17.38.2.2 initial_states()	175
17.38.2.3 reset()	176
17.39 epiworld::epimodels::ModelSEIRMixing< TSeq > Class Template Reference	176
17.39.1 Constructor & Destructor Documentation	177
17.39.1.1 ModelSEIRMixing() [1/2]	178
17.39.1.2 ModelSEIRMixing() [2/2]	178
17.39.2 Member Function Documentation	179
17.39.2.1 clone_ptr()	179
17.39.2.2 initial_states()	179
17.39.2.3 reset()	180
17.40 ModelSEIRMixing< TSeq > Class Template Reference	180
17.40.1 Constructor & Destructor Documentation	181
17.40.1.1 ModelSEIRMixing() [1/2]	181
17.40.1.2 ModelSEIRMixing() [2/2]	182
17.40.2 Member Function Documentation	182
17.40.2.1 clone_ptr()	183
17.40.2.2 initial_states()	183
17.40.2.3 reset()	183
17.41 epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq > Class Template Reference	184
17.41.1 Detailed Description	186
17.41.2 Constructor & Destructor Documentation	186
17.41.2.1 ModelSEIRMixingQuarantine() [1/2]	187
17.41.2.2 ModelSEIRMixingQuarantine() [2/2]	188
17.41.3 Member Function Documentation	189
17.41.3.1 clone_ptr()	189
17.41.3.2 get_agent_quarantine_triggered()	189
17.41.3.3 get_contact_matrix()	190
17.41.3.4 get_isolation_willingness()	190
17.41.3.5 get_quarantine_willingness()	190

17.41.3.6 initial_states()	190
17.41.3.7 run()	191
17.41.3.8 set_contact_matrix()	191
17.42 ModelSEIRMixingQuarantine< TSeq > Class Template Reference	192
17.42.1 Detailed Description	194
17.42.2 Constructor & Destructor Documentation	194
17.42.2.1 ModelSEIRMixingQuarantine() [1/2]	195
17.42.2.2 ModelSEIRMixingQuarantine() [2/2]	196
17.42.3 Member Function Documentation	197
17.42.3.1 clone_ptr()	197
17.42.3.2 get_agent_quarantine_triggered()	197
17.42.3.3 get_contact_matrix()	198
17.42.3.4 get_isolation_willingness()	198
17.42.3.5 get_quarantine_willingness()	198
17.42.3.6 initial_states()	198
17.42.3.7 run()	199
17.42.3.8 set_contact_matrix()	199
17.43 epiworld::epimodels::ModelSIR< TSeq > Class Template Reference	200
17.43.1 Detailed Description	201
17.43.2 Member Function Documentation	201
17.43.2.1 initial_states()	201
17.44 ModelSIR< TSeq > Class Template Reference	201
17.44.1 Detailed Description	202
17.44.2 Member Function Documentation	203
17.44.2.1 initial_states()	203
17.45 epiworld::epimodels::ModelSIRCONN< TSeq > Class Template Reference	204
17.45.1 Constructor & Destructor Documentation	205
17.45.1.1 ModelSIRCONN()	205
17.45.2 Member Function Documentation	205
17.45.2.1 clone_ptr()	206
17.45.2.2 get_n_infected()	206
17.45.2.3 initial_states()	206
17.45.2.4 reset()	206
17.46 ModelSIRCONN< TSeq > Class Template Reference	207
17.46.1 Constructor & Destructor Documentation	208
17.46.1.1 ModelSIRCONN()	208
17.46.2 Member Function Documentation	209
17.46.2.1 clone_ptr()	209
17.46.2.2 get_n_infected()	209
17.46.2.3 initial_states()	209
17.46.2.4 reset()	210
17.47 epiworld::epimodels::ModelSIRD< TSeq > Class Template Reference	210

17.47.1 Detailed Description	211
17.47.2 Constructor & Destructor Documentation	211
17.47.2.1 ModelSIRD()	212
17.47.3 Member Function Documentation	213
17.47.3.1 initial_states()	213
17.48 ModelSIRD< TSeq > Class Template Reference	213
17.48.1 Detailed Description	214
17.48.2 Constructor & Destructor Documentation	215
17.48.2.1 ModelSIRD()	215
17.48.3 Member Function Documentation	215
17.48.3.1 initial_states()	215
17.49 epiworld::epimodels::ModelSIRDCONN< TSeq > Class Template Reference	216
17.49.1 Constructor & Destructor Documentation	217
17.49.1.1 ModelSIRDCONN()	217
17.49.2 Member Function Documentation	218
17.49.2.1 clone_ptr()	218
17.49.2.2 reset()	218
17.50 ModelSIRDCONN< TSeq > Class Template Reference	219
17.50.1 Constructor & Destructor Documentation	220
17.50.1.1 ModelSIRDCONN()	220
17.50.2 Member Function Documentation	220
17.50.2.1 clone_ptr()	220
17.50.2.2 reset()	221
17.51 epiworld::epimodels::ModelSIRLogit< TSeq > Class Template Reference	221
17.51.1 Detailed Description	222
17.51.2 Constructor & Destructor Documentation	223
17.51.2.1 ModelSIRLogit()	223
17.51.3 Member Function Documentation	224
17.51.3.1 clone_ptr()	224
17.51.3.2 reset()	224
17.52 ModelSIRLogit< TSeq > Class Template Reference	225
17.52.1 Detailed Description	226
17.52.2 Constructor & Destructor Documentation	226
17.52.2.1 ModelSIRLogit()	226
17.52.3 Member Function Documentation	227
17.52.3.1 clone_ptr()	227
17.52.3.2 reset()	227
17.53 epiworld::epimodels::ModelSIRMixing< TSeq > Class Template Reference	228
17.53.1 Constructor & Destructor Documentation	229
17.53.1.1 ModelSIRMixing() [1/2]	229
17.53.1.2 ModelSIRMixing() [2/2]	230
17.53.2 Member Function Documentation	230

17.53.2.1 clone_ptr()	231
17.53.2.2 initial_states()	231
17.53.2.3 reset()	231
17.54 ModelSIRMixing< TSeq > Class Template Reference	232
17.54.1 Constructor & Destructor Documentation	233
17.54.1.1 ModelSIRMixing() [1/2]	233
17.54.1.2 ModelSIRMixing() [2/2]	234
17.54.2 Member Function Documentation	234
17.54.2.1 clone_ptr()	234
17.54.2.2 initial_states()	234
17.54.2.3 reset()	235
17.55 epiworld::epimodels::ModelSIS< TSeq > Class Template Reference	235
17.55.1 Detailed Description	237
17.56 ModelSIS< TSeq > Class Template Reference	238
17.56.1 Detailed Description	239
17.57 epiworld::epimodels::ModelSISD< TSeq > Class Template Reference	239
17.57.1 Detailed Description	240
17.58 ModelSISD< TSeq > Class Template Reference	241
17.58.1 Detailed Description	242
17.59 epiworld::epimodels::ModelSURV< TSeq > Class Template Reference	242
17.59.1 Member Function Documentation	244
17.59.1.1 reset()	244
17.60 ModelSURV< TSeq > Class Template Reference	245
17.60.1 Member Function Documentation	246
17.60.1.1 reset()	247
17.61 Network< Nettype, Nodetype, Edgetype > Class Template Reference	247
17.62 epiworld::PersonTools< TSeq > Class Template Reference	247
17.63 PersonTools< TSeq > Class Template Reference	248
17.64 epiworld::Progress Class Reference	248
17.64.1 Detailed Description	248
17.65 Progress Class Reference	248
17.65.1 Detailed Description	248
17.66 epiworld::Queue< TSeq > Class Template Reference	249
17.66.1 Detailed Description	249
17.67 Queue< TSeq > Class Template Reference	249
17.67.1 Detailed Description	250
17.68 RandGraph Class Reference	250
17.69 epiworld::SAMPLETYPE Class Reference	251
17.70 SAMPLETYPE Class Reference	251
17.71 epiworld::Tool< TSeq > Class Template Reference	251
17.71.1 Detailed Description	252
17.72 Tool< TSeq > Class Template Reference	252

17.72.1 Detailed Description	254
17.73 <code>epiworld::ToolFunctions< TSeq ></code> Class Template Reference	254
17.73.1 Detailed Description	254
17.74 <code>ToolFunctions< TSeq ></code> Class Template Reference	255
17.74.1 Detailed Description	255
17.75 <code>epiworld::Tools< TSeq ></code> Class Template Reference	255
17.75.1 Detailed Description	255
17.76 <code>Tools< TSeq ></code> Class Template Reference	256
17.76.1 Detailed Description	256
17.77 <code>epiworld::Tools_const< TSeq ></code> Class Template Reference	257
17.77.1 Detailed Description	257
17.78 <code>Tools_const< TSeq ></code> Class Template Reference	257
17.78.1 Detailed Description	258
17.79 <code>epiworld::UserData< TSeq ></code> Class Template Reference	258
17.79.1 Detailed Description	259
17.79.2 Constructor & Destructor Documentation	260
17.79.2.1 <code>UserData()</code>	260
17.80 <code>UserData< TSeq ></code> Class Template Reference	260
17.80.1 Detailed Description	261
17.80.2 Constructor & Destructor Documentation	261
17.80.2.1 <code>UserData()</code>	261
17.81 <code>epiworld::vecHasher< T ></code> Struct Template Reference	262
17.81.1 Detailed Description	262
17.82 <code>vecHasher< T ></code> Struct Template Reference	262
17.82.1 Detailed Description	262
17.83 <code>epiworld::Virus< TSeq ></code> Class Template Reference	263
17.83.1 Detailed Description	264
17.84 <code>Virus< TSeq ></code> Class Template Reference	265
17.84.1 Detailed Description	267
17.85 <code>epiworld::Viruses< TSeq ></code> Class Template Reference	267
17.85.1 Detailed Description	267
17.86 <code>Viruses< TSeq ></code> Class Template Reference	268
17.86.1 Detailed Description	268
17.87 <code>epiworld::Viruses_const< TSeq ></code> Class Template Reference	269
17.87.1 Detailed Description	269
17.88 <code>Viruses_const< TSeq ></code> Class Template Reference	269
17.88.1 Detailed Description	270
17.89 <code>epiworld::VirusFunctions< TSeq ></code> Class Template Reference	270
17.90 <code>VirusFunctions< TSeq ></code> Class Template Reference	271
18 File Documentation	273
18.1 <code>include/epiworld/agent-meat-state.hpp</code> File Reference	273

18.1.1 Detailed Description	274
18.2 include/epiworld/models/measlesmixing.hpp File Reference	275
18.2.1 Detailed Description	275
18.2.2 Macro Definition Documentation	275
18.2.2.1 GET_MODEL	276
18.2.2.2 SAMPLE_FROM_PROBS	276
18.3 include/epiworld/models/seirmixingquarantine.hpp File Reference	276
18.3.1 Detailed Description	277
18.3.2 Macro Definition Documentation	277
18.3.2.1 GET_MODEL	277
18.3.2.2 SAMPLE_FROM_PROBS	277
Index	279

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

ModelMeaslesMixingRiskQuarantine

A comprehensive epidemiological model that extends measles transmission modeling with risk-stratified quarantine strategies.

5.1 Overview

`ModelMeaslesMixingRiskQuarantine` implements a measles transmission model based on the [ModelMeaslesMixing](#) framework, with enhanced quarantine policies that vary based on exposure risk levels. This allows for targeted public health interventions that can optimize resource allocation and epidemic control.

5.1.1 Key Features

- **Measles-specific disease progression:** Susceptible → Exposed → Prodromal → Rash → Recovered
- **Population mixing:** Uses contact matrices to model heterogeneous mixing between population groups
- **Risk-stratified quarantine:** Three-tier system (high/medium/low risk) with customizable durations
- **Enhanced detection:** Detection rate parameter that activates during quarantine periods
- **Contact tracing:** Comprehensive contact tracing with configurable success rates
- **Entity-based risk assessment:** Risk levels determined by shared entity membership
- **Vaccination effects:** Vaccine distribution with configurable efficacy

5.2 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** - Individuals who have recovered and gained immunity

5.3 Risk Classification System

When the quarantine process is triggered by a detected case, contacts are classified into three risk levels:

5.3.1 High Risk

- **Definition:** Unvaccinated agents who share entity membership with the case that triggered quarantine
- **Rationale:** Highest transmission risk due to close, prolonged contact within same household/workplace/school
- **Default quarantine duration:** 21 days (configurable)

5.3.2 Medium Risk

- **Definition:** Unvaccinated agents who had contact with infected individuals but don't share entity membership
- **Rationale:** Moderate transmission risk from community contact
- **Default quarantine duration:** 14 days (configurable)

5.3.3 Low Risk

- **Definition:** All other unvaccinated agents
- **Rationale:** Lowest transmission risk, potential for community spread
- **Default quarantine duration:** 7 days (configurable)

Note: Vaccinated agents (those with tools) are not subject to quarantine regardless of contact patterns.

5.4 Enhanced Detection

The model includes an enhanced detection mechanism that operates during active quarantine periods:

- **Detection rate quarantine:** Additional parameter specifying detection probability for prodromal individuals during active quarantine
- **Mechanism:** When any quarantine process is active, prodromal individuals have an additional chance of being detected and moved to quarantined prodromal state
- **Purpose:** Models increased surveillance and testing during outbreak response

5.5 Parameters

5.5.1 Disease Parameters

- **Contact rate:** Average number of contacts per step
- **Transmission rate:** Probability of transmission per contact
- **Incubation period:** Average time from exposure to becoming infectious
- **Prodromal period:** Average duration of infectious period
- **Rash period:** Average duration of rash symptoms
- **Hospitalization rate:** Probability of requiring hospitalization
- **Hospitalization period:** Average duration of hospital stay

5.5.2 Detection and Isolation

- **Days undetected:** Average time before rash cases are detected
- **Isolation period:** Duration of isolation for detected cases
- **Isolation willingness:** Proportion willing to self-isolate when detected
- **Detection rate quarantine:** Detection rate during active quarantine periods

5.5.3 Risk-based Quarantine

- **Quarantine period high:** Duration for high-risk contacts (days)
- **Quarantine period medium:** Duration for medium-risk contacts (days)
- **Quarantine period low:** Duration for low-risk contacts (days)
- **Quarantine willingness:** Proportion willing to comply with quarantine

5.5.4 Contact Tracing

- **Contact tracing success rate:** Probability of successfully identifying contacts
- **Contact tracing days prior:** Number of days prior to detection for tracing

5.5.5 Vaccination

- **Vaccination rate:** Proportion of agents initially vaccinated
- **Vax efficacy:** Vaccine effectiveness in preventing infection
- **Vax improved recovery:** Enhanced recovery rate for vaccinated individuals

5.6 Usage Example

```
#include <epiworld/epiworld.hpp>
// Contact matrix for 3 population groups
std::vector<double> contact_matrix = {
    0.8, 0.1, 0.1, // Group 1 mixing
    0.1, 0.8, 0.1, // Group 2 mixing
    0.1, 0.1, 0.8 // Group 3 mixing
};
// Create model with risk-based quarantine
epimodels::ModelMeaslesMixingRiskQuarantine<> model(
    1000, // Population size
    0.01, // Initial prevalence
    2.0, // Contact rate
    0.2, // Transmission rate
    0.9, // Vaccination efficacy
    0.3, // Vaccine recovery enhancement
    7.0, // Incubation period
    4.0, // Prodromal period
    5.0, // Rash period
    contact_matrix, // Contact matrix
    0.2, // Hospitalization rate
    7.0, // Hospitalization duration
    3.0, // Days undetected
    21, // Quarantine period high risk
    14, // Quarantine period medium risk
    7, // Quarantine period low risk
    0.8, // Quarantine willingness
    0.8, // Isolation willingness
    4, // Isolation period
    0.1, // Proportion vaccinated
    0.15, // Detection rate during quarantine
    1.0, // Contact tracing success rate
    4u // Contact tracing days prior
);
// Add population entities
model.add_entity(Entity<>("Households", dist_factory<>(0, 400)));
model.add_entity(Entity<>("Schools", dist_factory<>(400, 700)));
model.add_entity(Entity<>("Workplaces", dist_factory<>(700, 1000)));
// Set initial conditions
model.initial_states({1.0, 0.0}); // All infected start as exposed
// Run simulation
model.run(60, 123);
model.print();
```

5.7 Differences from Other Models

5.7.1 vs. ModelMeaslesMixing

- **Risk-stratified quarantine:** Three different quarantine durations based on exposure risk
- **Enhanced detection:** Additional detection during active quarantine periods
- **Entity-based risk assessment:** Risk levels determined by shared entity membership
- **Flexible quarantine policies:** Each risk level can have different quarantine duration or be disabled

5.7.2 vs. ModelMeaslesSchool

- **Population mixing:** Supports contact matrices for heterogeneous mixing
- **Risk stratification:** Multiple quarantine strategies rather than uniform approach
- **Enhanced contact tracing:** Risk-based contact management
- **Scalability:** Supports multiple population entities with different mixing patterns

5.8 Testing

The model includes comprehensive tests that verify:

- Correct transition probabilities between all 13 states
- Proper risk level assignment (high/medium/low)
- Enhanced detection mechanism during quarantine periods
- Different quarantine durations for different risk levels
- Comparison of uniform vs. risk-stratified quarantine strategies
- Population mixing functionality with multiple entities

See `tests/20a-measles-mixing-risk-quarantine.cpp` for test examples.

5.9 Applications

This model is particularly useful for:

1. **Outbreak response planning:** Comparing different quarantine strategies
2. **Resource optimization:** Allocating quarantine resources based on transmission risk
3. **Policy evaluation:** Assessing effectiveness of risk-stratified interventions
4. **Contact tracing optimization:** Understanding impact of enhanced detection during outbreaks
5. **Vaccination strategy:** Evaluating how vaccination coverage affects quarantine effectiveness

5.10 Implementation Notes

- Risk levels are assigned dynamically when quarantine is triggered
- Vaccinated agents are excluded from quarantine regardless of contact patterns
- Enhanced detection only operates when at least one quarantine process is active
- Quarantine durations can be set to -1 to disable quarantine for specific risk levels
- The model maintains backward compatibility with uniform quarantine by setting all periods equal

Chapter 6

epiworld c++ template library

6.1 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.

6.2 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.

6.3 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;
}
```

6.4 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.

6.4.1 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)")
```

6.4.2 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 7

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
)
```

7.1 Compartmental Models

7.1.1 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")

```

7.1.2 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")

```

7.2 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
```

7.2.1 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.

7.2.2 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")
```

7.3 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.

7.3.1 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.

7.3.2 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)))
```

7.3.3 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 8

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 9

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 ( $N(i, t, s, k|v)$ )  $k$ .

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 10

# Mixing probabilities in connected model

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

### 10.1 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.

## 10.2 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 11

## EPI Simulator

### 11.1 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.

### 11.2 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.

### 11.3 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.

### 11.4 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

## 11.5 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.

### 11.5.1 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 12

# Namespace Index

### 12.1 Namespace List

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

<a href="#">epiworld::sampler</a>	Functions for sampling viruses . . . . .	<a href="#">43</a>
<a href="#">sampler</a>	Functions for sampling viruses . . . . .	<a href="#">46</a>



## Chapter 13

# Hierarchical Index

### 13.1 Class Hierarchy

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

AdjList . . . . .	51
epiworld::AdjList . . . . .	52
Agent< TSeq > . . . . .	54
epiworld::Agent< TSeq > . . . . .	59
Agent< EPI_DEFAULT_TSEQ > . . . . .	54
AgentsSample< TSeq > . . . . .	62
epiworld::AgentsSample< TSeq > . . . . .	64
DataBase< TSeq > . . . . .	65
epiworld::DataBase< TSeq > . . . . .	71
Entities< TSeq > . . . . .	75
epiworld::Entities< TSeq > . . . . .	76
Entities_const< TSeq > . . . . .	77
epiworld::Entities_const< TSeq > . . . . .	78
Entity< TSeq > . . . . .	79
epiworld::Entity< TSeq > . . . . .	80
Entity< EPI_DEFAULT_TSEQ > . . . . .	79
epiworld::Event< TSeq > . . . . .	82
Event< TSeq > . . . . .	83
epiworld::GlobalEvent< TSeq > . . . . .	85
GlobalEvent< TSeq > . . . . .	87
epiworld::LFMCMC< TData > . . . . .	88
LFMCMC< TData > . . . . .	89
epiworld::Model< TSeq > . . . . .	91
Model< TSeq > . . . . .	106
ModelMeaslesSchool< TSeq > . . . . .	148
epiworld::Model< EPI_DEFAULT_TSEQ > . . . . .	91
ModelDiffNet< TSeq > . . . . .	125
ModelMeaslesMixing< TSeq > . . . . .	135
ModelSEIR< TSeq > . . . . .	155
ModelSEIRCONN< TSeq > . . . . .	161
ModelSEIRD< TSeq > . . . . .	167
ModelSEIRDCONN< TSeq > . . . . .	173
ModelSEIRMixing< TSeq > . . . . .	180
ModelSEIRMixingQuarantine< TSeq > . . . . .	192
ModelSIR< TSeq > . . . . .	201

ModelSIRCONN< TSeq > . . . . .	207
ModelSIRD< TSeq > . . . . .	213
ModelSIRDCONN< TSeq > . . . . .	219
ModelSIRLogit< TSeq > . . . . .	225
ModelSIRMixing< TSeq > . . . . .	232
ModelSIS< TSeq > . . . . .	238
ModelSISD< TSeq > . . . . .	241
ModelSURV< TSeq > . . . . .	245
epiworld::epimodels::ModelDiffNet< TSeq > . . . . .	123
epiworld::epimodels::ModelMeaslesMixing< TSeq > . . . . .	126
epiworld::epimodels::ModelMeaslesSchool< TSeq > . . . . .	143
epiworld::epimodels::ModelSEIR< TSeq > . . . . .	152
epiworld::epimodels::ModelSEIRCONN< TSeq > . . . . .	158
epiworld::epimodels::ModelSEIRD< TSeq > . . . . .	164
epiworld::epimodels::ModelSEIRDCONN< TSeq > . . . . .	170
epiworld::epimodels::ModelSEIRMixing< TSeq > . . . . .	176
epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq > . . . . .	184
epiworld::epimodels::ModelSIR< TSeq > . . . . .	200
epiworld::epimodels::ModelSIRCONN< TSeq > . . . . .	204
epiworld::epimodels::ModelSIRD< TSeq > . . . . .	210
epiworld::epimodels::ModelSIRDCONN< TSeq > . . . . .	216
epiworld::epimodels::ModelSIRLogit< TSeq > . . . . .	221
epiworld::epimodels::ModelSIRMixing< TSeq > . . . . .	228
epiworld::epimodels::ModelSIS< TSeq > . . . . .	235
epiworld::epimodels::ModelSISD< TSeq > . . . . .	239
epiworld::epimodels::ModelSURV< TSeq > . . . . .	242
epiworld::ModelDiagram . . . . .	123
ModelDiagram . . . . .	123
Network< Nettype, Nodetype, Edgetype > . . . . .	247
epiworld::PersonTools< TSeq > . . . . .	247
PersonTools< TSeq > . . . . .	248
epiworld::Progress . . . . .	248
Progress . . . . .	248
epiworld::Queue< TSeq > . . . . .	249
Queue< TSeq > . . . . .	249
RandGraph . . . . .	250
epiworld::SAMPLETYPE . . . . .	251
SAMPLETYPE . . . . .	251
epiworld::Tool< TSeq > . . . . .	251
Tool< TSeq > . . . . .	252
epiworld::ToolFunctions< TSeq > . . . . .	254
ToolFunctions< TSeq > . . . . .	255
epiworld::Tools< TSeq > . . . . .	255
Tools< TSeq > . . . . .	256
epiworld::Tools_const< TSeq > . . . . .	257
Tools_const< TSeq > . . . . .	257
epiworld::UserData< TSeq > . . . . .	258
UserData< TSeq > . . . . .	260
epiworld::vecHasher< T > . . . . .	262
vecHasher< T > . . . . .	262
epiworld::Virus< TSeq > . . . . .	263
Virus< TSeq > . . . . .	265
epiworld::Viruses< TSeq > . . . . .	267
Viruses< TSeq > . . . . .	268
epiworld::Viruses_const< TSeq > . . . . .	269
Viruses_const< TSeq > . . . . .	269
epiworld::VirusFunctions< TSeq > . . . . .	270
VirusFunctions< TSeq > . . . . .	271

## Chapter 14

# Class Index

### 14.1 Class List

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

<a href="#">AdjList</a> . . . . .	51
<a href="#">epiworld::AdjList</a> . . . . .	52
<a href="#">Agent&lt; TSeq &gt;</a>	
<a href="#">Agent</a> (agents) . . . . .	54
<a href="#">epiworld::Agent&lt; TSeq &gt;</a>	
<a href="#">Agent</a> (agents) . . . . .	59
<a href="#">AgentsSample&lt; TSeq &gt;</a>	
Sample of agents . . . . .	62
<a href="#">epiworld::AgentsSample&lt; TSeq &gt;</a>	
Sample of agents . . . . .	64
<a href="#">DataBase&lt; TSeq &gt;</a>	
Statistical data about the process . . . . .	65
<a href="#">epiworld::DataBase&lt; TSeq &gt;</a>	
Statistical data about the process . . . . .	71
<a href="#">Entities&lt; TSeq &gt;</a>	
Set of <a href="#">Entities</a> (useful for building iterators) . . . . .	75
<a href="#">epiworld::Entities&lt; TSeq &gt;</a>	
Set of <a href="#">Entities</a> (useful for building iterators) . . . . .	76
<a href="#">Entities_const&lt; TSeq &gt;</a>	
Set of <a href="#">Entities</a> (const) (useful for iterators) . . . . .	77
<a href="#">epiworld::Entities_const&lt; TSeq &gt;</a>	
Set of <a href="#">Entities</a> (const) (useful for iterators) . . . . .	78
<a href="#">Entity&lt; TSeq &gt;</a> . . . . .	79
<a href="#">epiworld::Entity&lt; TSeq &gt;</a> . . . . .	80
<a href="#">epiworld::Event&lt; TSeq &gt;</a>	
<a href="#">Event</a> data for update an agent . . . . .	82
<a href="#">Event&lt; TSeq &gt;</a>	
<a href="#">Event</a> data for update an agent . . . . .	83
<a href="#">epiworld::GlobalEvent&lt; TSeq &gt;</a>	
Template for a Global <a href="#">Event</a> . . . . .	85
<a href="#">GlobalEvent&lt; TSeq &gt;</a>	
Template for a Global <a href="#">Event</a> . . . . .	87
<a href="#">epiworld::LFMCMC&lt; TData &gt;</a>	
Likelihood-Free Markov Chain Monte Carlo . . . . .	88
<a href="#">LFMCMC&lt; TData &gt;</a>	
Likelihood-Free Markov Chain Monte Carlo . . . . .	89

<a href="#">epiworld::Model&lt; TSeq &gt;</a>	
Core class of epiworld	91
<a href="#">Model&lt; TSeq &gt;</a>	
Core class of epiworld	106
<a href="#">epiworld::ModelDiagram</a>	123
<a href="#">ModelDiagram</a>	123
<a href="#">epiworld::epimodels::ModelDiffNet&lt; TSeq &gt;</a>	
Template for a <a href="#">Network Diffusion Model</a>	123
<a href="#">ModelDiffNet&lt; TSeq &gt;</a>	
Template for a <a href="#">Network Diffusion Model</a>	125
<a href="#">epiworld::epimodels::ModelMeaslesMixing&lt; TSeq &gt;</a>	
Measles model with population mixing, quarantine, and contact tracing	126
<a href="#">ModelMeaslesMixing&lt; TSeq &gt;</a>	
Measles model with population mixing, quarantine, and contact tracing	135
<a href="#">epiworld::epimodels::ModelMeaslesSchool&lt; TSeq &gt;</a>	
Template for a Measles model with quarantine	143
<a href="#">ModelMeaslesSchool&lt; TSeq &gt;</a>	
Template for a Measles model with quarantine	148
<a href="#">epiworld::epimodels::ModelSEIR&lt; TSeq &gt;</a>	
Template for a Susceptible-Exposed-Infected-Removed (SEIR) model	152
<a href="#">ModelSEIR&lt; TSeq &gt;</a>	
Template for a Susceptible-Exposed-Infected-Removed (SEIR) model	155
<a href="#">epiworld::epimodels::ModelSEIRCONN&lt; TSeq &gt;</a>	158
<a href="#">ModelSEIRCONN&lt; TSeq &gt;</a>	161
<a href="#">epiworld::epimodels::ModelSEIRD&lt; TSeq &gt;</a>	
Template for a Susceptible-Exposed-Infected-Removed-Deceased (SEIRD) model	164
<a href="#">ModelSEIRD&lt; TSeq &gt;</a>	
Template for a Susceptible-Exposed-Infected-Removed-Deceased (SEIRD) model	167
<a href="#">epiworld::epimodels::ModelSEIRDCONN&lt; TSeq &gt;</a>	170
<a href="#">ModelSEIRDCONN&lt; TSeq &gt;</a>	173
<a href="#">epiworld::epimodels::ModelSEIRMixing&lt; TSeq &gt;</a>	176
<a href="#">ModelSEIRMixing&lt; TSeq &gt;</a>	180
<a href="#">epiworld::epimodels::ModelSEIRMixingQuarantine&lt; TSeq &gt;</a>	
SEIR model with mixing, quarantine, and contact tracing	184
<a href="#">ModelSEIRMixingQuarantine&lt; TSeq &gt;</a>	
SEIR model with mixing, quarantine, and contact tracing	192
<a href="#">epiworld::epimodels::ModelSIR&lt; TSeq &gt;</a>	
Template for a Susceptible-Infected-Removed (SIR) model	200
<a href="#">ModelSIR&lt; TSeq &gt;</a>	
Template for a Susceptible-Infected-Removed (SIR) model	201
<a href="#">epiworld::epimodels::ModelSIRCONN&lt; TSeq &gt;</a>	204
<a href="#">ModelSIRCONN&lt; TSeq &gt;</a>	207
<a href="#">epiworld::epimodels::ModelSIRD&lt; TSeq &gt;</a>	
Template for a Susceptible-Infected-Removed-Deceased (SIRD) model	210
<a href="#">ModelSIRD&lt; TSeq &gt;</a>	
Template for a Susceptible-Infected-Removed-Deceased (SIRD) model	213
<a href="#">epiworld::epimodels::ModelSIRDCONN&lt; TSeq &gt;</a>	216
<a href="#">ModelSIRDCONN&lt; TSeq &gt;</a>	219
<a href="#">epiworld::epimodels::ModelSIRLogit&lt; TSeq &gt;</a>	
Template for a Susceptible-Infected-Removed (SIR) model	221
<a href="#">ModelSIRLogit&lt; TSeq &gt;</a>	
Template for a Susceptible-Infected-Removed (SIR) model	225
<a href="#">epiworld::epimodels::ModelSIRMixing&lt; TSeq &gt;</a>	228
<a href="#">ModelSIRMixing&lt; TSeq &gt;</a>	232
<a href="#">epiworld::epimodels::ModelSIS&lt; TSeq &gt;</a>	
Template for a Susceptible-Infected-Susceptible (SIS) model	235
<a href="#">ModelSIS&lt; TSeq &gt;</a>	
Template for a Susceptible-Infected-Susceptible (SIS) model	238

<a href="#">epiworld::epimodels::ModelSISD&lt; TSeq &gt;</a>	
Template for a Susceptible-Infected-Susceptible-Deceased (SISD) model	239
<a href="#">ModelSISD&lt; TSeq &gt;</a>	
Template for a Susceptible-Infected-Susceptible-Deceased (SISD) model	241
<a href="#">epiworld::epimodels::ModelSURV&lt; TSeq &gt;</a>	242
<a href="#">ModelSURV&lt; TSeq &gt;</a>	245
<a href="#">Network&lt; Nettype, Nodetype, Edgetype &gt;</a>	247
<a href="#">epiworld::PersonTools&lt; TSeq &gt;</a>	247
<a href="#">PersonTools&lt; TSeq &gt;</a>	248
<a href="#">epiworld::Progress</a>	
A simple progress bar	248
<a href="#">Progress</a>	
A simple progress bar	248
<a href="#">epiworld::Queue&lt; TSeq &gt;</a>	
Controls which agents are verified at each step	249
<a href="#">Queue&lt; TSeq &gt;</a>	
Controls which agents are verified at each step	249
<a href="#">RandGraph</a>	250
<a href="#">epiworld::SAMPLETYPE</a>	251
<a href="#">SAMPLETYPE</a>	251
<a href="#">epiworld::Tool&lt; TSeq &gt;</a>	
Tools for defending the agent against the virus	251
<a href="#">Tool&lt; TSeq &gt;</a>	
Tools for defending the agent against the virus	252
<a href="#">epiworld::ToolFunctions&lt; TSeq &gt;</a>	
Helper class to store the functions avoiding multiple shared_pointers (we have only one for the four of these)	254
<a href="#">ToolFunctions&lt; TSeq &gt;</a>	
Helper class to store the functions avoiding multiple shared_pointers (we have only one for the four of these)	255
<a href="#">epiworld::Tools&lt; TSeq &gt;</a>	
Set of tools (useful for building iterators)	255
<a href="#">Tools&lt; TSeq &gt;</a>	
Set of tools (useful for building iterators)	256
<a href="#">epiworld::Tools_const&lt; TSeq &gt;</a>	
Set of <a href="#">Tools</a> (const) (useful for iterators)	257
<a href="#">Tools_const&lt; TSeq &gt;</a>	
Set of <a href="#">Tools</a> (const) (useful for iterators)	257
<a href="#">epiworld::UserData&lt; TSeq &gt;</a>	
Personalized data by the user	258
<a href="#">UserData&lt; TSeq &gt;</a>	
Personalized data by the user	260
<a href="#">epiworld::vecHasher&lt; T &gt;</a>	
Vector hasher	262
<a href="#">vecHasher&lt; T &gt;</a>	
Vector hasher	262
<a href="#">epiworld::Virus&lt; TSeq &gt;</a>	
Virus	263
<a href="#">Virus&lt; TSeq &gt;</a>	
Virus	265
<a href="#">epiworld::Viruses&lt; TSeq &gt;</a>	
Set of viruses (useful for building iterators)	267
<a href="#">Viruses&lt; TSeq &gt;</a>	
Set of viruses (useful for building iterators)	268
<a href="#">epiworld::Viruses_const&lt; TSeq &gt;</a>	
Set of <a href="#">Viruses</a> (const) (useful for iterators)	269
<a href="#">Viruses_const&lt; TSeq &gt;</a>	
Set of <a href="#">Viruses</a> (const) (useful for iterators)	269

<a href="#">epiworld::VirusFunctions&lt; TSeq &gt;</a>	<a href="#">270</a>
<a href="#">VirusFunctions&lt; TSeq &gt;</a>	<a href="#">271</a>



## Chapter 15

# File Index

### 15.1 File List

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

<b>epiworld.hpp</b>	??
include/epiworld/ <b>adjlist-bones.hpp</b>	??
include/epiworld/ <b>adjlist-meat.hpp</b>	??
include/epiworld/ <b>agent-bones.hpp</b>	??
include/epiworld/ <b>agent-events-meat.hpp</b>	??
include/epiworld/ <b>agent-meat-state.hpp</b>	??
Sampling functions are getting big, so we keep them in a separate file	<a href="#">273</a>
include/epiworld/ <b>agent-meat-virus-sampling.hpp</b>	??
include/epiworld/ <b>agent-meat.hpp</b>	??
include/epiworld/ <b>agentssample-bones.hpp</b>	??
include/epiworld/ <b>config.hpp</b>	??
include/epiworld/ <b>database-bones.hpp</b>	??
include/epiworld/ <b>database-meat.hpp</b>	??
include/epiworld/ <b>entities-bones.hpp</b>	??
include/epiworld/ <b>entity-bones.hpp</b>	??
include/epiworld/ <b>entity-distribute-meat.hpp</b>	??
include/epiworld/ <b>entity-meat.hpp</b>	??
include/epiworld/ <b>epiworld-macros.hpp</b>	??
include/epiworld/ <b>epiworld.hpp</b>	??
include/epiworld/ <b>globalevent-bones.hpp</b>	??
include/epiworld/ <b>globalevent-meat.hpp</b>	??
include/epiworld/ <b>misc.hpp</b>	??
include/epiworld/ <b>model-bones.hpp</b>	??
include/epiworld/ <b>model-meat-print.hpp</b>	??
include/epiworld/ <b>model-meat.hpp</b>	??
include/epiworld/ <b>modeldiagram-bones.hpp</b>	??
include/epiworld/ <b>modeldiagram-meat.hpp</b>	??
include/epiworld/ <b>network-bones.hpp</b>	??
include/epiworld/ <b>progress.hpp</b>	??
include/epiworld/ <b>queue-bones.hpp</b>	??
include/epiworld/ <b>randgraph.hpp</b>	??
include/epiworld/ <b>random_graph.hpp</b>	??
include/epiworld/ <b>seq_processing.hpp</b>	??
include/epiworld/ <b>tool-bones.hpp</b>	??
include/epiworld/ <b>tool-distribute-meat.hpp</b>	??

include/epiworld/ <b>tool-meat.hpp</b>	??
include/epiworld/ <b>tools-bones.hpp</b>	??
include/epiworld/ <b>userdata-bones.hpp</b>	??
include/epiworld/ <b>userdata-meat.hpp</b>	??
include/epiworld/ <b>virus-bones.hpp</b>	??
include/epiworld/ <b>virus-distribute-meat.hpp</b>	??
include/epiworld/ <b>virus-meat.hpp</b>	??
include/epiworld/ <b>viruses-bones.hpp</b>	??
include/epiworld/math/ <b>distributions.hpp</b>	??
include/epiworld/math/ <b>lfmcmc.hpp</b>	??
include/epiworld/math/lfmcmc/ <b>lfmcmc-bones.hpp</b>	??
include/epiworld/math/lfmcmc/ <b>lfmcmc-meat-print.hpp</b>	??
include/epiworld/math/lfmcmc/ <b>lfmcmc-meat.hpp</b>	??
include/epiworld/models/ <b>diffnet.hpp</b>	??
include/epiworld/models/ <b>globalevents.hpp</b>	??
include/epiworld/models/ <b>init-functions.hpp</b>	??
include/epiworld/models/ <a href="#">measlesmixing.hpp</a>	
Template for a Measles model with population mixing, quarantine, and contact tracing	275
include/epiworld/models/ <b>measlesquarantine.hpp</b>	??
include/epiworld/models/ <b>models.hpp</b>	??
include/epiworld/models/ <b>seir.hpp</b>	??
include/epiworld/models/ <b>seirconnected.hpp</b>	??
include/epiworld/models/ <b>seird.hpp</b>	??
include/epiworld/models/ <b>seirdconnected.hpp</b>	??
include/epiworld/models/ <b>seirmixing.hpp</b>	??
include/epiworld/models/ <a href="#">seirmixingquarantine.hpp</a>	
Template for a Susceptible-Exposed-Infected-Removed (SEIR) model with mixing, quarantine, and contact tracing	276
include/epiworld/models/ <b>sir.hpp</b>	??
include/epiworld/models/ <b>sirconnected.hpp</b>	??
include/epiworld/models/ <b>sird.hpp</b>	??
include/epiworld/models/ <b>sirdconnected.hpp</b>	??
include/epiworld/models/ <b>sirlogit.hpp</b>	??
include/epiworld/models/ <b>sirmixing.hpp</b>	??
include/epiworld/models/ <b>sis.hpp</b>	??
include/epiworld/models/ <b>sisd.hpp</b>	??
include/epiworld/models/ <b>surveillance.hpp</b>	??
tests/ <b>tests.hpp</b>	??

## Chapter 16

# Namespace Documentation

### 16.1 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)*

#### 16.1.1 Detailed Description

Functions for sampling viruses.

#### 16.1.2 Function Documentation

##### 16.1.2.1 `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;

## 16.1.2.2 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;

## 16.1.2.3 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;

## 16.2 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)*

### 16.2.1 Detailed Description

Functions for sampling viruses.

### 16.2.2 Function Documentation

#### 16.2.2.1 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;

## 16.2.2.2 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;

## 16.2.2.3 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 17

# Class Documentation

### 17.1 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.*

#### 17.1.1 Constructor & Destructor Documentation

##### 17.1.1.1 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

## 17.1.2 Member Function Documentation

### 17.1.2.1 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>	<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 files:

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

## 17.2 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.*

## 17.2.1 Constructor & Destructor Documentation

### 17.2.1.1 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

## 17.2.2 Member Function Documentation

### 17.2.2.1 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`

## 17.3 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	<a href="#">Tool</a> to add
virus	<a href="#">Virus</a> 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)

## 17.3.1 Detailed Description

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

[Agent](#) (agents)

Template Parameters

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

## 17.3.2 Member Function Documentation

### 17.3.2.1 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&amp;

**17.3.2.2 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>	

**17.3.3 Friends And Related Function Documentation****17.3.3.1 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

## 17.4 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><a href="#">Tool</a> to add</i>
virus	<i><a href="#">Virus</a> 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)

### 17.4.1 Detailed Description

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

[Agent](#) (agents)

Template Parameters

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

### 17.4.2 Member Function Documentation

#### 17.4.2.1 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&

### 17.4.2.2 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>	

## 17.4.3 Friends And Related Function Documentation

### 17.4.3.1 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

## 17.5 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

### 17.5.1 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>	
-------------	--

### 17.5.2 Constructor & Destructor Documentation

#### 17.5.2.1 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

## 17.6 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

### 17.6.1 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>	
-------------	--

## 17.6.2 Constructor &amp; Destructor Documentation

## 17.6.2.1 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

## 17.7 DataBase&lt; TSeq &gt; 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)

### 17.7.1 Detailed Description

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

Statistical data about the process.

#### Template Parameters

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

## 17.7.2 Member Function Documentation

### 17.7.2.1 `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.

### 17.7.2.2 `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  $R_0$  (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

### 17.7.2.3 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 >

### 17.7.2.4 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>	

### 17.7.2.5 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,

#### 17.7.2.6 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,

#### 17.7.2.7 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

#### 17.7.2.8 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

## 17.8 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)

## 17.8.1 Detailed Description

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

Statistical data about the process.

### Template Parameters

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

## 17.8.2 Member Function Documentation

### 17.8.2.1 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.

### 17.8.2.2 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

#### 17.8.2.3 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 >`

#### 17.8.2.4 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>	

## 17.8.2.5 operator==( )

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

&lt; Date of the transmission eve

&lt; Id of the sour

&lt; Id of the targ

&lt; Id of the varia

&lt; Date when the source acquired the varia

## 17.8.2.6 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

## 17.9 Entities&lt; TSeq &gt; Class Template Reference

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

#include &lt;entities-bones.hpp&gt;

## 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 >

### 17.9.1 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

## 17.10 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 >

### 17.10.1 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

## 17.11 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 >

### 17.11.1 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

## 17.12 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 >

#### 17.12.1 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

## 17.13 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< size\_t >::iterator **begin** ()
- std::vector< size\_t >::iterator **end** ()
- std::vector< size\_t >::const\_iterator **begin** () const
- std::vector< size\_t >::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)

## 17.13.1 Constructor & Destructor Documentation

### 17.13.1.1 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.

## 17.13.2 Friends And Related Function Documentation

### 17.13.2.1 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

## 17.14 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< size\_t >::iterator **begin** ()
- std::vector< size\_t >::iterator **end** ()
- std::vector< size\_t >::const\_iterator **begin** () const
- std::vector< size\_t >::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)

## 17.14.1 Constructor & Destructor Documentation

### 17.14.1.1 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.

## 17.14.2 Friends And Related Function Documentation

### 17.14.2.1 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

## 17.15 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**

### 17.15.1 Detailed Description

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

Event data for update an agent.

## Template Parameters

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

## 17.15.2 Constructor &amp; Destructor Documentation

## 17.15.2.1 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>	<a href="#">Agent</a> over who the action will happen
<i>virus_</i>	<a href="#">Virus</a> to add
<i>tool_</i>	<a href="#">Tool</a> 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:

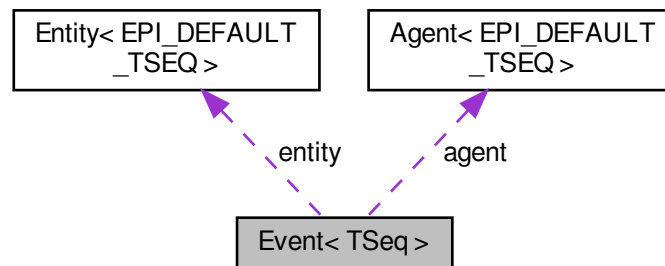
- epiworld.hpp

## 17.16 Event&lt; TSeq &gt; 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**

### 17.16.1 Detailed Description

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

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

Template Parameters

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

## 17.16.2 Constructor & Destructor Documentation

### 17.16.2.1 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>	<a href="#">Agent</a> over who the action will happen
<i>virus_</i>	<a href="#">Virus</a> to add
<i>tool_</i>	<a href="#">Tool</a> 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

## 17.17 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

### 17.17.1 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.

### 17.17.2 Constructor & Destructor Documentation

#### 17.17.2.1 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 <a href="#">Model</a> <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

## 17.18 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

### 17.18.1 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.

### 17.18.2 Constructor & Destructor Documentation

#### 17.18.2.1 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 <a href="#">Model</a> <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

## 17.19 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)

**17.19.1 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

**17.20 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)`

## 17.20.1 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:

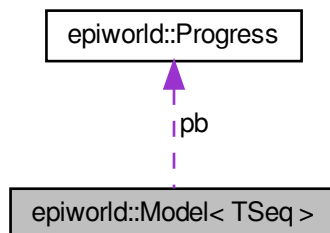
- include/epiworld/math/lfmcmc/lfmcmc-bones.hpp
- include/epiworld/math/lfmcmc/lfmcmc-meat-print.hpp
- include/epiworld/math/lfmcmc/lfmcmc-meat.hpp

## 17.21 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><a href="#">Virus</a> to be added</i>
t	<i><a href="#">Tool</a> 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><a href="#">AdjList</a> 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** ()

### 17.21.1 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.)
-------------	-----------------------------------------------------------------------------------------------------------------------------------------

### 17.21.2 Member Function Documentation

#### 17.21.2.1 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.

### 17.21.2.2 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 >](#).

### 17.21.2.3 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.

### 17.21.2.4 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>	<a href="#">Agent</a> over which the action will be called
<i>virus_</i>	<a href="#">Virus</a> pointer included in the action
<i>tool_</i>	<a href="#">Tool</a> pointer included in the action
<i>entity_</i>	<a href="#">Entity</a> 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.

## 17.21.2.5 events\_run()

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

Executes the stored action.

## Parameters

<i>model↔ _</i>	<a href="#">Model</a> over which it will be executed.
---------------------	-------------------------------------------------------

## 17.21.2.6 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.

### 17.21.2.7 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 >](#).

### 17.21.2.8 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
--------------	---------------------------------

### 17.21.2.9 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.

#### 17.21.2.10 set\_name()

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

Set the name object.

#### Parameters

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

#### 17.21.2.11 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>	

## 17.21.3 Member Data Documentation

### 17.21.3.1 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

### 17.21.3.2 rbinomd

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

**Initial value:**

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

### 17.21.3.3 rexp

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

**Initial value:**

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

### 17.21.3.4 rgammad

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

**Initial value:**

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



### 17.21.3.5 rgeomd

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

**Initial value:**

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

### 17.21.3.6 rlognormald

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

**Initial value:**

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

### 17.21.3.7 rnbinomd

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

**Initial value:**

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

### 17.21.3.8 rnormd

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

**Initial value:**

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

### 17.21.3.9 rpoissd

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

**Initial value:**

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

**17.21.3.10 runifd**

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

**Initial value:**

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

**17.21.3.11 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:

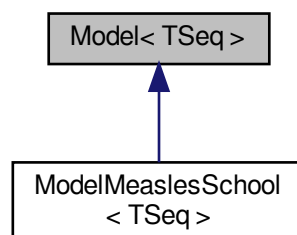
- epiworld.hpp

**17.22 Model< TSeq > Class Template Reference**

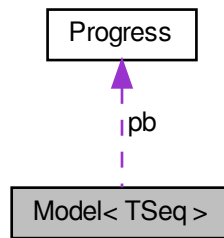
Core class of epiworld.

```
#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\_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	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><a href="#">AdjList</a> 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 ()`

### 17.22.1 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.)
-------------	-----------------------------------------------------------------------------------------------------------------------------------------

## 17.22.2 Member Function Documentation

## 17.22.2.1 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.

## 17.22.2.2 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 >](#).

## 17.22.2.3 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.

### 17.22.2.4 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>	<a href="#">Agent</a> over which the action will be called
<i>virus_</i>	<a href="#">Virus</a> pointer included in the action
<i>tool_</i>	<a href="#">Tool</a> pointer included in the action
<i>entity_</i>	<a href="#">Entity</a> 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.

### 17.22.2.5 events\_run()

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

Executes the stored action.

## Parameters

<i>model</i> ↔	<a href="#">Model</a> over which it will be executed.
—	

**17.22.2.6 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.

**17.22.2.7 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 >](#).

**17.22.2.8 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
--------------	---------------------------------

**17.22.2.9 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.

**17.22.2.10 set\_name()**

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

Set the name object.

**Parameters**

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

### 17.22.2.11 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

## 17.22.3 Member Data Documentation

### 17.22.3.1 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

### 17.22.3.2 rbinomd

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

#### Initial value:

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



### 17.22.3.3 rexp

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

**Initial value:**

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

### 17.22.3.4 rgammad

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

**Initial value:**

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

### 17.22.3.5 rgeomd

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

**Initial value:**

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

### 17.22.3.6 rlognormald

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

**Initial value:**

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

### 17.22.3.7 rnbinomd

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

**Initial value:**

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

### 17.22.3.8 rnormd

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

**Initial value:**

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

### 17.22.3.9 rpoissd

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

**Initial value:**

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

### 17.22.3.10 runifd

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

**Initial value:**

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

### 17.22.3.11 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

## 17.23 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

## 17.24 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:

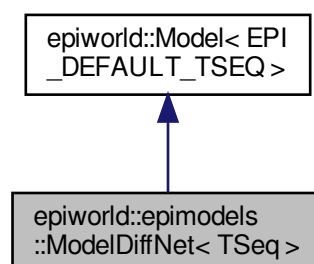
- include/epiworld/modeldiagram-bones.hpp
- include/epiworld/modeldiagram-meat.hpp

## 17.25 epiworld::epimodels::ModelDiffNet< TSeq > Class Template Reference

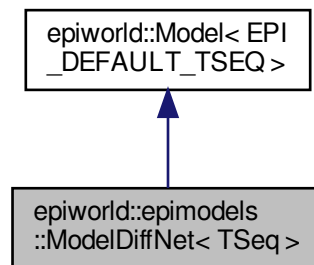
Template for a [Network](#) Diffusion [Model](#).

```
#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

### 17.25.1 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:

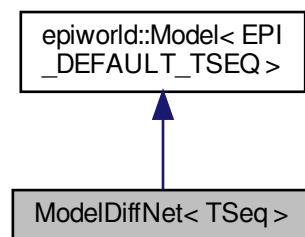
- epiworld.hpp

## 17.26 ModelDiffNet< TSeq > Class Template Reference

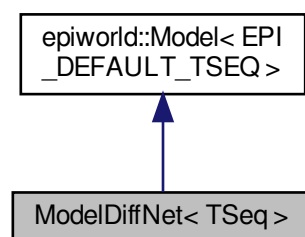
Template for a [Network](#) Diffusion [Model](#).

```
#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

### 17.26.1 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:

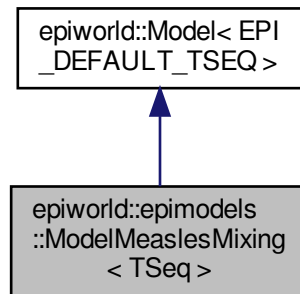
- include/epiworld/models/diffnet.hpp

## 17.27 epiworld::epimodels::ModelMeaslesMixing< TSeq > Class Template Reference

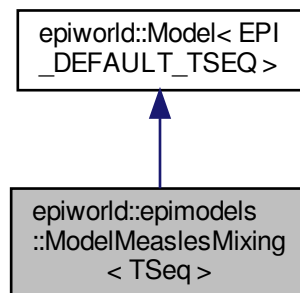
Measles model with population mixing, quarantine, and contact tracing.

```
#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

### 17.27.1 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> )
-------------------	----------------------------------------------------------------------

## 17.27.2 Constructor & Destructor Documentation

### 17.27.2.1 `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 <a href="#">ModelMeaslesMixing</a> 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 <a href="#">ModelMeaslesMixing&lt;TSeq&gt;</a> 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

17.27.2.2 `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).

### 17.27.3 Member Function Documentation

#### 17.27.3.1 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 >](#).

#### 17.27.3.2 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

### 17.27.3.3 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

### 17.27.3.4 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

### 17.27.3.5 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

### 17.27.3.6 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"> <li>• [0]: The proportion of initially infected individuals who start in the exposed state.</li> <li>• [1]: The proportion of initially non-infected individuals who have recovered (immune).</li> </ul>
<i>queue_</i>	Optional vector for queuing specifications (default: empty).

Reimplemented from [epiworld::Model< EPI\\_DEFAULT\\_TSEQ >](#).

**17.27.3.7 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 >](#).

**17.27.3.8 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:

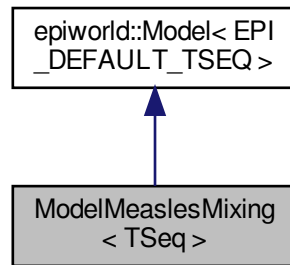
- epiworld.hpp

## 17.28 ModelMeaslesMixing< TSeq > Class Template Reference

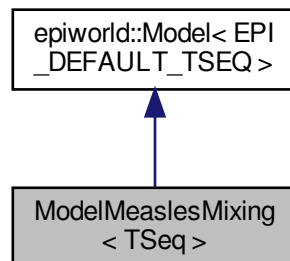
Measles model with population mixing, quarantine, and contact tracing.

```
#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

### 17.28.1 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)
-------------	--------------------------------------------------------

## 17.28.2 Constructor &amp; Destructor Documentation

## 17.28.2.1 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 <a href="#">ModelMeaslesMixing</a> 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

## 17.28.2.2 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).

## 17.28.3 Member Function Documentation

### 17.28.3.1 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 >](#).

**17.28.3.2 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

**17.28.3.3 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

**17.28.3.4 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

### 17.28.3.5 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

### 17.28.3.6 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"> <li>[0]: The proportion of initially infected individuals who start in the exposed state.</li> <li>[1]: The proportion of initially non-infected individuals who have recovered (immune).</li> </ul>
<i>queue_</i>	Optional vector for queuing specifications (default: empty).

Reimplemented from [epiworld::Model< EPI\\_DEFAULT\\_TSEQ >](#).

### 17.28.3.7 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 >](#).

**17.28.3.8 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:

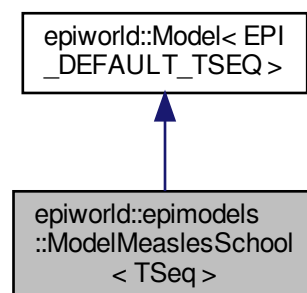
- [include/epiworld/models/measlesmixing.hpp](#)

## 17.29 epiworld::epimodels::ModelMeaslesSchool< TSeq > Class Template Reference

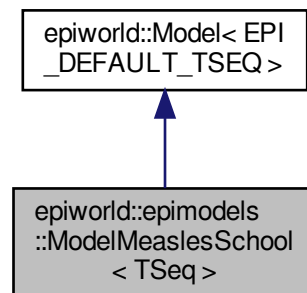
Template for a Measles model with quarantine.

```
#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 constexpr `epiworld_fast_uint` **SUSCEPTIBLE** = 0u
- static constexpr `epiworld_fast_uint` **EXPOSED** = 1u
- static constexpr `epiworld_fast_uint` **PRODROMAL** = 2u
- static constexpr `epiworld_fast_uint` **RASH** = 3u
- static constexpr `epiworld_fast_uint` **ISOLATED** = 4u
- static constexpr `epiworld_fast_uint` **ISOLATED\_RECOVERED** = 5u
- static constexpr `epiworld_fast_uint` **DETECTED\_HOSPITALIZED** = 6u
- static constexpr `epiworld_fast_uint` **QUARANTINED\_EXPOSED** = 7u
- static constexpr `epiworld_fast_uint` **QUARANTINED\_SUSCEPTIBLE** = 8u
- static constexpr `epiworld_fast_uint` **QUARANTINED\_PRODROMAL** = 9u
- static constexpr `epiworld_fast_uint` **QUARANTINED\_RECOVERED** = 10u
- static constexpr `epiworld_fast_uint` **HOSPITALIZED** = 11u
- static constexpr `epiworld_fast_uint` **RECOVERED** = 12u

## Additional Inherited Members

### 17.29.1 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.

### 17.29.2 Constructor & Destructor Documentation

#### 17.29.2.1 `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.

## 17.29.3 Member Function Documentation

### 17.29.3.1 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 >](#).

### 17.29.3.2 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.

### 17.29.3.3 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:

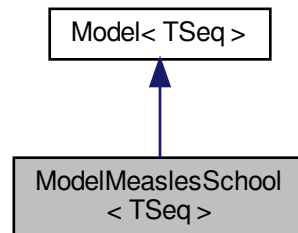
- epiworld.hpp

## 17.30 ModelMeaslesSchool< TSeq > Class Template Reference

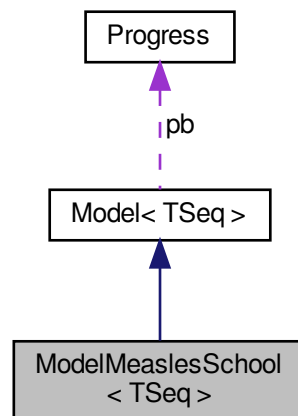
Template for a Measles model with quarantine.

```
#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 constexpr](#) [epiworld\\_fast\\_uint](#) **SUSCEPTIBLE** = 0u
- [static constexpr](#) [epiworld\\_fast\\_uint](#) **EXPOSED** = 1u
- [static constexpr](#) [epiworld\\_fast\\_uint](#) **PRODROMAL** = 2u
- [static constexpr](#) [epiworld\\_fast\\_uint](#) **RASH** = 3u
- [static constexpr](#) [epiworld\\_fast\\_uint](#) **ISOLATED** = 4u
- [static constexpr](#) [epiworld\\_fast\\_uint](#) **ISOLATED\_RECOVERED** = 5u
- [static constexpr](#) [epiworld\\_fast\\_uint](#) **DETECTED\_HOSPITALIZED** = 6u
- [static constexpr](#) [epiworld\\_fast\\_uint](#) **QUARANTINED\_EXPOSED** = 7u
- [static constexpr](#) [epiworld\\_fast\\_uint](#) **QUARANTINED\_SUSCEPTIBLE** = 8u
- [static constexpr](#) [epiworld\\_fast\\_uint](#) **QUARANTINED\_PRODROMAL** = 9u
- [static constexpr](#) [epiworld\\_fast\\_uint](#) **QUARANTINED\_RECOVERED** = 10u
- [static constexpr](#) [epiworld\\_fast\\_uint](#) **HOSPITALIZED** = 11u
- [static constexpr](#) [epiworld\\_fast\\_uint](#) **RECOVERED** = 12u

## Additional Inherited Members

### 17.30.1 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.

## 17.30.2 Constructor & Destructor Documentation

### 17.30.2.1 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.

### 17.30.3 Member Function Documentation

#### 17.30.3.1 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 >](#).

#### 17.30.3.2 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.

### 17.30.3.3 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:

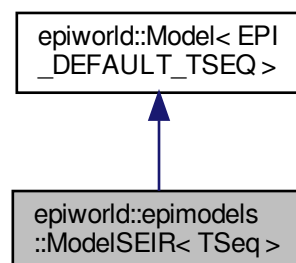
- `include/epiworld/models/measlesquarantine.hpp`

## 17.31 epiworld::epimodels::ModelSEIR< TSeq > Class Template Reference

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

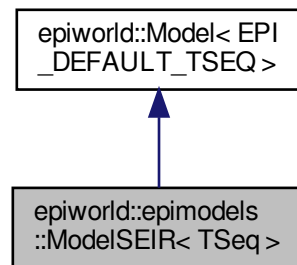
```
#include <epiworld.hpp>
```

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

### 17.31.1 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.

## 17.31.2 Member Function Documentation

### 17.31.2.1 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"> <li>• 0: Proportion of non-infected agents who are removed.</li> <li>• 1: Proportion of exposed agents to be set as infected.</li> </ul>

Reimplemented from [epiworld::Model< EPI\\_DEFAULT\\_TSEQ >](#).

## 17.31.3 Member Data Documentation

### 17.31.3.1 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;
}
```

## 17.31.3.2 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:

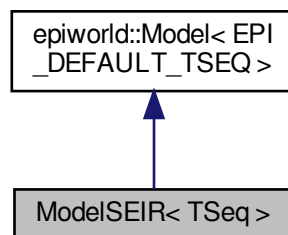
- epiworld.hpp

## 17.32 ModelSEIR&lt; TSeq &gt; Class Template Reference

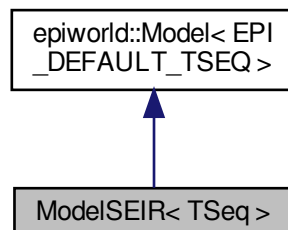
Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.

```
#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

### 17.32.1 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.

### 17.32.2 Member Function Documentation

### 17.32.2.1 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"> <li>• 0: Proportion of non-infected agents who are removed.</li> <li>• 1: Proportion of exposed agents to be set as infected.</li> </ul>

Reimplemented from [epiworld::Model< EPI\\_DEFAULT\\_TSEQ >](#).

## 17.32.3 Member Data Documentation

### 17.32.3.1 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;
}
```

### 17.32.3.2 update\_infected\_seir

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

#### Initial value:

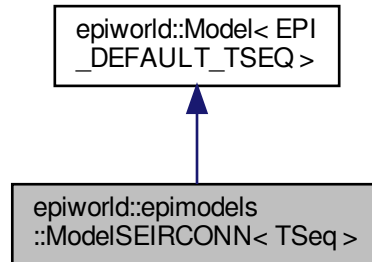
```
= [] (
 epiworld::Agent<TSeq> * p,
 epiworld::Model<TSeq> * m
) -> void {
 if (m->runif() < (m->par("Recovery rate")))
 p->rm_virus(m);
 return;
}
```

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

- include/epiworld/models/seir.hpp

## 17.33 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

### 17.33.1 Constructor & Destructor Documentation

#### 17.33.1.1 ModelSEIRCONN()

```
template<typename TSeq >
ModelSEIRCONN< TSeq >::ModelSEIRCONN (
 ModelSEIRCONN< TSeq > & model,
 const std::string & vname,
 ewiworld_fast_uint n,
 ewiworld_double prevalence,
 ewiworld_double contact_rate,
 ewiworld_double transmission_rate,
 ewiworld_double avg_incubation_days,
 ewiworld_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

### 17.33.2 Member Function Documentation

### 17.33.2.1 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 >](#).

### 17.33.2.2 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"> <li>The proportion of non-infected individuals who have recovered.</li> </ul>

Reimplemented from [epiworld::Model< EPI\\_DEFAULT\\_TSEQ >](#).

### 17.33.2.3 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 >](#).

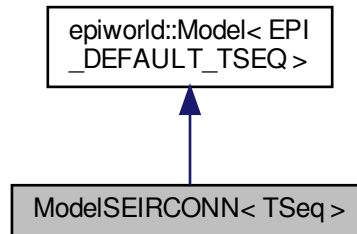
The documentation for this class was generated from the following file:

- `epiworld.hpp`

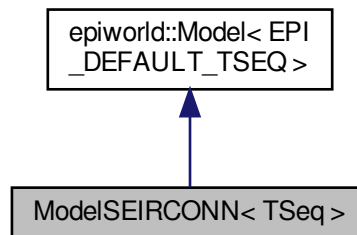


## 17.34 ModelSEIRCONN< TSeq > Class Template Reference

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

### 17.34.1 Constructor & Destructor Documentation

#### 17.34.1.1 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

### 17.34.2 Member Function Documentation

### 17.34.2.1 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 >](#).

### 17.34.2.2 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"> <li>The proportion of non-infected individuals who have recovered.</li> </ul>

Reimplemented from [epiworld::Model< EPI\\_DEFAULT\\_TSEQ >](#).

### 17.34.2.3 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:

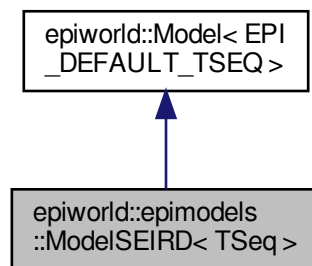
- `include/epiworld/models/seirconnected.hpp`

## 17.35 epiworld::epimodels::ModelSEIRD< TSeq > Class Template Reference

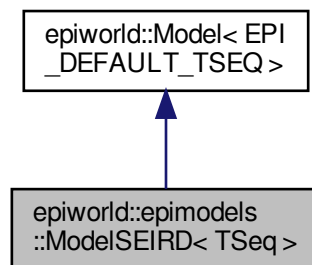
Template for a Susceptible-Exposed-Infected-Removed-Deceased (SEIRD) model.

```
#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

### 17.35.1 Detailed Description

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

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

### 17.35.2 Constructor & Destructor Documentation

#### 17.35.2.1 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.

**17.35.2.2 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.

**17.35.3 Member Data Documentation****17.35.3.1 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:

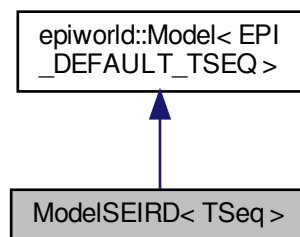
- `epiworld.hpp`

## 17.36 ModelSEIRD< TSeq > Class Template Reference

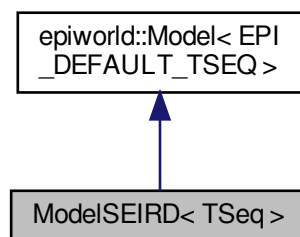
Template for a Susceptible-Exposed-Infected-Removed-Deceased (SEIRD) model.

```
#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

### 17.36.1 Detailed Description

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

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

### 17.36.2 Constructor & Destructor Documentation

#### 17.36.2.1 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.

## 17.36.2.2 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.

## 17.36.3 Member Data Documentation

## 17.36.3.1 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`

## 17.37 `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

### 17.37.1 Constructor & Destructor Documentation

#### 17.37.1.1 `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 <code>Model&lt;TSeq&gt;</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
<i>death_rate</i>	Probability of death

## 17.37.2 Member Function Documentation

### 17.37.2.1 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 >](#).

### 17.37.2.2 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"> <li>• 0: Proportion of non-infected agents who are removed.</li> <li>• 1: Proportion of exposed agents to be set as infected.</li> </ul>

Reimplemented from [epiworld::Model< EPI\\_DEFAULT\\_TSEQ >](#).

### 17.37.2.3 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`

## 17.38 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

### 17.38.1 Constructor & Destructor Documentation

#### 17.38.1.1 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

## 17.38.2 Member Function Documentation

## 17.38.2.1 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 >](#).

## 17.38.2.2 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"> <li>• 0: Proportion of non-infected agents who are removed.</li> <li>• 1: Proportion of exposed agents to be set as infected.</li> </ul>

Reimplemented from [epiworld::Model< EPI\\_DEFAULT\\_TSEQ >](#).

### 17.38.2.3 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`

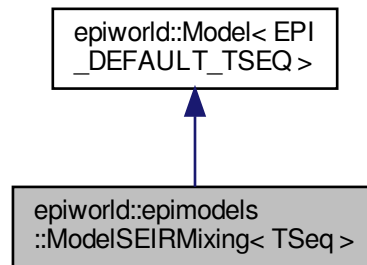
## 17.39 epiworld::epimodels::ModelSEIRMixing< TSeq > Class Template Reference

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

### 17.39.1 Constructor & Destructor Documentation

**17.39.1.1 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 <a href="#">ModelSEIRMixing</a> object.
<i>vname</i>	The name of the <a href="#">ModelSEIRMixing</a> 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

**17.39.1.2 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 <a href="#">ModelSEIRMixing</a> 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.

## 17.39.2 Member Function Documentation

17.39.2.1 `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 >](#).

17.39.2.2 `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"> <li>The proportion of non-infected individuals who have recovered.</li> </ul>

Reimplemented from [epiworld::Model< EPI\\_DEFAULT\\_TSEQ >](#).

### 17.39.2.3 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

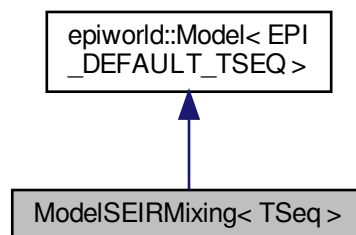
Reimplemented from `epiworld::Model< EPI_DEFAULT_TSEQ >`.

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

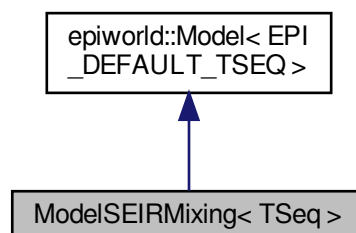
- `epiworld.hpp`

## 17.40 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

### 17.40.1 Constructor & Destructor Documentation

#### 17.40.1.1 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 <a href="#">ModelSEIRMixing</a> object.
<i>vname</i>	The name of the <a href="#">ModelSEIRMixing</a> 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&lt;TSeq&gt;</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

17.40.1.2 [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 <a href="#">ModelSEIRMixing</a> 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.

## 17.40.2 Member Function Documentation

### 17.40.2.1 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 >](#).

### 17.40.2.2 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"> <li>The proportion of non-infected individuals who have recovered.</li> </ul>

Reimplemented from [epiworld::Model< EPI\\_DEFAULT\\_TSEQ >](#).

### 17.40.2.3 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:

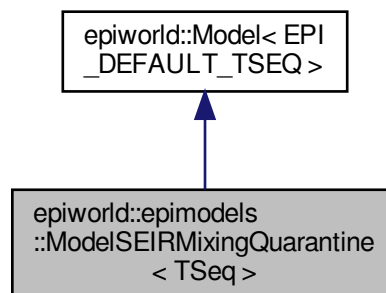
- `include/epiworld/models/seirmixing.hpp`

## 17.41 epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq > Class Template Reference

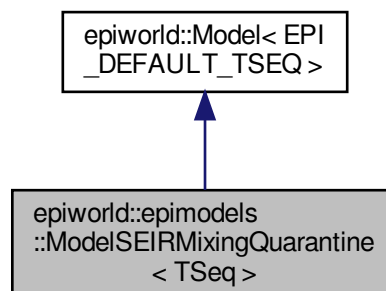
SEIR model with mixing, quarantine, and contact tracing.

```
#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

### 17.41.1 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)
-------------	--------------------------------------------------------

### 17.41.2 Constructor & Destructor Documentation

## 17.41.2.1 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 <a href="#">ModelSEIRMixingQuarantine</a> object.
<i>vname</i>	The name of the <a href="#">ModelSEIRMixingQuarantine</a> 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 <a href="#">ModelSEIRMixingQuarantine&lt;TSeq&gt;</a> 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

17.41.2.2 `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 <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

<code>recovery_rate</code>	The recovery rate of the disease in the model.
<code>contact_matrix</code>	The contact matrix between entities in the model.
<code>hospitalization_rate</code>	The rate at which infected individuals are hospitalized.
<code>hospitalization_period</code>	The average duration of hospitalization in days.
<code>days_undetected</code>	The average number of days an infected individual remains undetected.
<code>quarantine_period</code>	The duration of quarantine in days for exposed contacts.
<code>quarantine_willingness</code>	The proportion of individuals willing to comply with quarantine measures.
<code>isolation_willingness</code>	The proportion of individuals willing to self-isolate when detected.
<code>isolation_period</code>	The duration of isolation in days for detected infected individuals.
<code>contact_tracing_success_rate</code>	The probability of successfully identifying and tracing contacts (default: 1.0).
<code>contact_tracing_days_prior</code>	The number of days prior to detection for which contacts are traced (default: 4).

### 17.41.3 Member Function Documentation

#### 17.41.3.1 `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 >`.

#### 17.41.3.2 `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

### 17.41.3.3 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

### 17.41.3.4 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

### 17.41.3.5 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

### 17.41.3.6 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"> <li>• [0]: The proportion of initially infected individuals who start in the exposed state.</li> <li>• [1]: The proportion of initially non-infected individuals who have recovered (immune).</li> </ul>
<i>queue_</i>	Optional vector for queuing specifications (default: empty).

Reimplemented from [epiworld::Model< EPI\\_DEFAULT\\_TSEQ >](#).

## 17.41.3.7 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 >](#).

## 17.41.3.8 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:

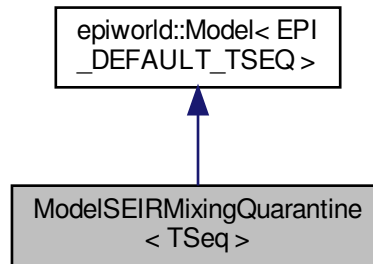
- epiworld.hpp

## 17.42 ModelSEIRMixingQuarantine< TSeq > Class Template Reference

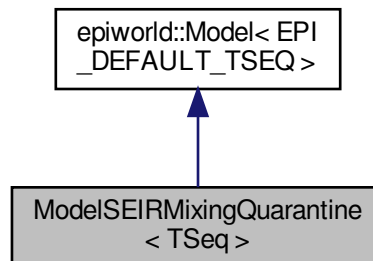
SEIR model with mixing, quarantine, and contact tracing.

```
#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

### 17.42.1 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)
-------------	--------------------------------------------------------

### 17.42.2 Constructor & Destructor Documentation

## 17.42.2.1 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 <a href="#">ModelSEIRMixingQuarantine</a> object.
<i>vname</i>	The name of the <a href="#">ModelSEIRMixingQuarantine</a> 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 <a href="#">ModelSEIRMixingQuarantine&lt;TSeq&gt;</a> 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

## 17.42.2.2 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 <a href="#">ModelSEIRMixingQuarantine</a> 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).

## 17.42.3 Member Function Documentation

### 17.42.3.1 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 >](#).

### 17.42.3.2 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

**17.42.3.3 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

**17.42.3.4 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

**17.42.3.5 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

**17.42.3.6 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"> <li>• [0]: The proportion of initially infected individuals who start in the exposed state.</li> <li>• [1]: The proportion of initially non-infected individuals who have recovered (immune).</li> </ul>
<i>queue_</i>	Optional vector for queuing specifications (default: empty).

Reimplemented from [epiworld::Model< EPI\\_DEFAULT\\_TSEQ >](#).

## 17.42.3.7 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 >](#).

## 17.42.3.8 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:

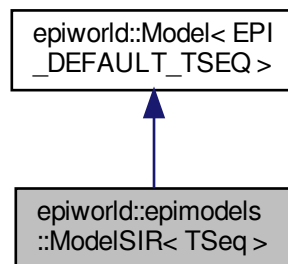
- include/epiworld/models/[seirmixingquarantine.hpp](#)

## 17.43 epiworld::epimodels::ModelSIR< TSeq > Class Template Reference

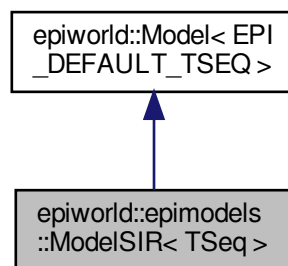
Template for a Susceptible-Infected-Removed (SIR) model.

```
#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

### 17.43.1 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

### 17.43.2 Member Function Documentation

#### 17.43.2.1 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"> <li>The proportion of non-infected individuals who have recovered.</li> </ul>

Reimplemented from [epiworld::Model< EPI\\_DEFAULT\\_TSEQ >](#).

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

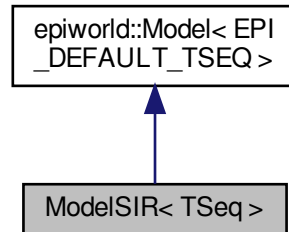
- epiworld.hpp

## 17.44 ModelSIR< TSeq > Class Template Reference

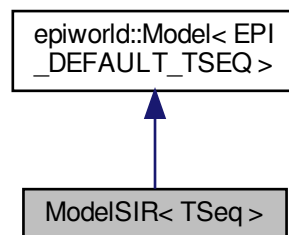
Template for a Susceptible-Infected-Removed (SIR) model.

```
#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

### 17.44.1 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

## 17.44.2 Member Function Documentation

## 17.44.2.1 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"> <li>The proportion of non-infected individuals who have recovered.</li> </ul>

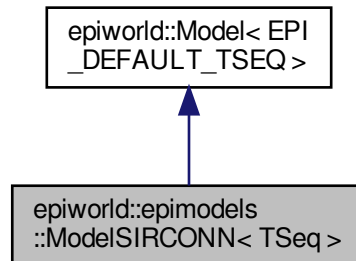
Reimplemented from [epiworld::Model< EPI\\_DEFAULT\\_TSEQ >](#).

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

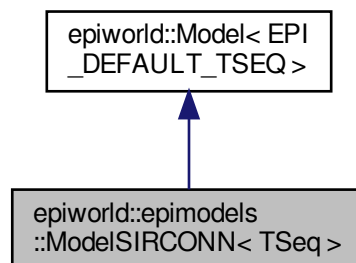
- include/epiworld/models/sir.hpp

## 17.45 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

### 17.45.1 Constructor & Destructor Documentation

#### 17.45.1.1 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

### 17.45.2 Member Function Documentation

### 17.45.2.1 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 >](#).

### 17.45.2.2 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> \* >

### 17.45.2.3 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"> <li>The proportion of non-infected individuals who have recovered.</li> </ul>

Reimplemented from [epiworld::Model< EPI\\_DEFAULT\\_TSEQ >](#).

### 17.45.2.4 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

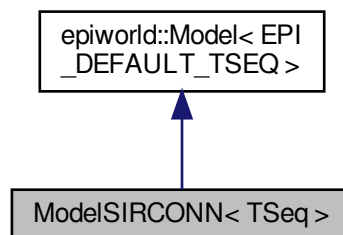
Reimplemented from [epiworld::Model< EPI\\_DEFAULT\\_TSEQ >](#).

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

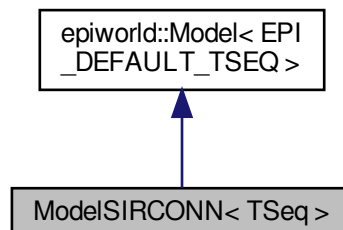
- `epiworld.hpp`

## 17.46 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

### 17.46.1 Constructor & Destructor Documentation

#### 17.46.1.1 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



## 17.46.2 Member Function Documentation

### 17.46.2.1 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 >](#).

### 17.46.2.2 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> \* >

### 17.46.2.3 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"> <li>The proportion of non-infected individuals who have recovered.</li> </ul>

Reimplemented from [epiworld::Model< EPI\\_DEFAULT\\_TSEQ >](#).

#### 17.46.2.4 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:

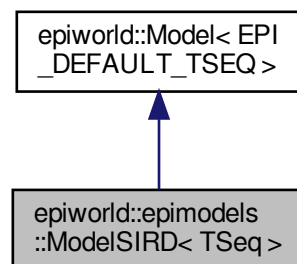
- `include/epiworld/models/sirconnected.hpp`

## 17.47 epiworld::epimodels::ModelSIRD< TSeq > Class Template Reference

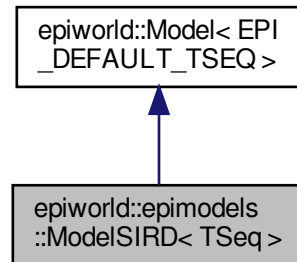
Template for a Susceptible-Infected-Removed-Deceased (SIRD) model.

```
#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

### 17.47.1 Detailed Description

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

Template for a Susceptible-Infected-Removed-Deceased (SIRD) model.

### 17.47.2 Constructor & Destructor Documentation

### 17.47.2.1 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.

## 17.47.3 Member Function Documentation

## 17.47.3.1 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"> <li>• The proportion of non-infected individuals who have recovered.</li> <li>• The proportion of non-infected individuals who have died.</li> </ul>

Reimplemented from [epiworld::Model< EPI\\_DEFAULT\\_TSEQ >](#).

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

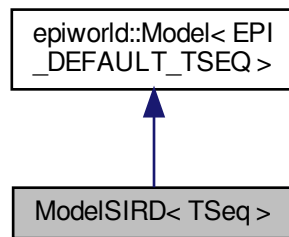
- [epiworld.hpp](#)

## 17.48 ModelSIRD&lt; TSeq &gt; Class Template Reference

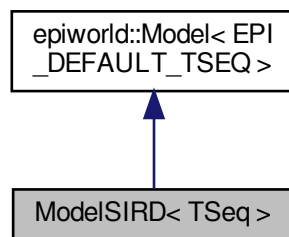
Template for a Susceptible-Infected-Removed-Deceased (SIRD) model.

```
#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

### 17.48.1 Detailed Description

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

Template for a Susceptible-Infected-Removed-Deceased (SIRD) model.

## 17.48.2 Constructor & Destructor Documentation

### 17.48.2.1 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.

## 17.48.3 Member Function Documentation

### 17.48.3.1 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"> <li>The proportion of non-infected individuals who have recovered.</li> <li>The proportion of non-infected individuals who have died.</li> </ul>

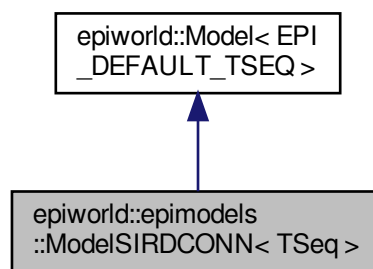
Reimplemented from [epiworld::Model< EPI\\_DEFAULT\\_TSEQ >](#).

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

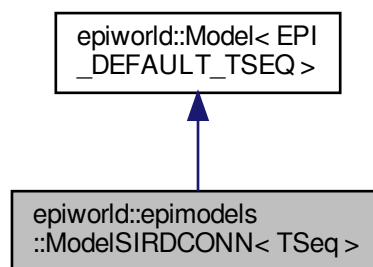
- include/epiworld/models/sird.hpp

## 17.49 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, 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

### 17.49.1 Constructor & Destructor Documentation

#### 17.49.1.1 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

## 17.49.2 Member Function Documentation

### 17.49.2.1 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 >](#).

### 17.49.2.2 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

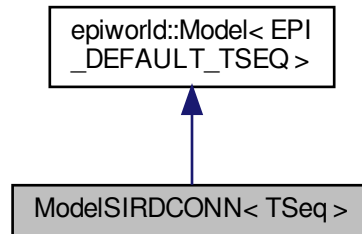
Reimplemented from [epiworld::Model< EPI\\_DEFAULT\\_TSEQ >](#).

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

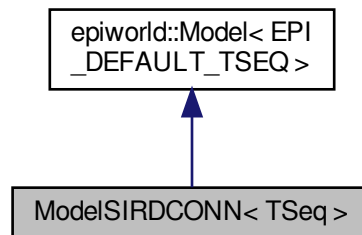
- `epiworld.hpp`

## 17.50 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

### 17.50.1 Constructor & Destructor Documentation

#### 17.50.1.1 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

### 17.50.2 Member Function Documentation

#### 17.50.2.1 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 >](#).

**17.50.2.2 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:

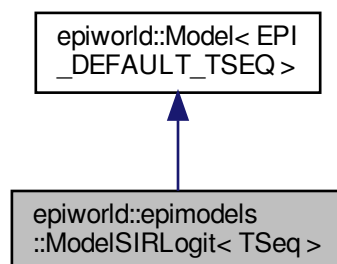
- include/epiworld/models/sirdconnected.hpp

## 17.51 epiworld::epimodels::ModelSIRLogit< TSeq > Class Template Reference

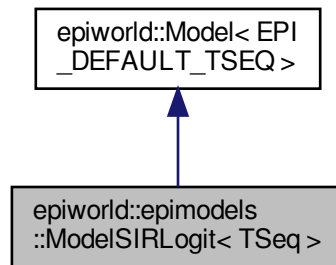
Template for a Susceptible-Infected-Removed (SIR) model.

```
#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

### 17.51.1 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  $\beta_0$  is the exposure coefficient and  $E_i$  is the exposure number,  $\beta_i$  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  $\beta_i$  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> )
-------------	---------------------------------------------------------------------------------

## 17.51.2 Constructor & Destructor Documentation

### 17.51.2.1 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>ncoenf_infect</i>	Unsigned int. Number of infection coefficients.
<i>ncoenf_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

### 17.51.3 Member Function Documentation

#### 17.51.3.1 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 >](#).

#### 17.51.3.2 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`

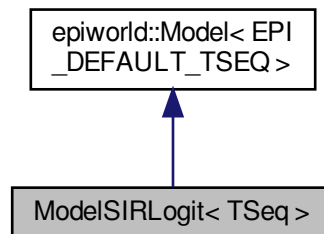


## 17.52 ModelSIRLogit< TSeq > Class Template Reference

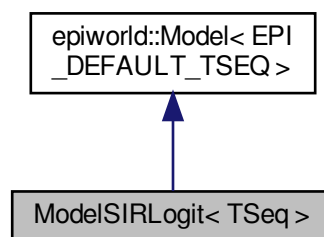
Template for a Susceptible-Infected-Removed (SIR) model.

```
#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

### 17.52.1 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  $\beta_0$  is the exposure coefficient and  $E_i$  is the exposure number,  $\beta_i$  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  $\beta_i$  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> )
-------------	---------------------------------------------------------------------------------

### 17.52.2 Constructor & Destructor Documentation

#### 17.52.2.1 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>ncoef_infect</i>	Unsigned int. Number of infection coefficients.
<i>ncoef_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

## 17.52.3 Member Function Documentation

### 17.52.3.1 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 >](#).

### 17.52.3.2 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

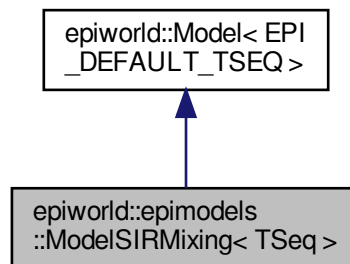
Reimplemented from [epiworld::Model< EPI\\_DEFAULT\\_TSEQ >](#).

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

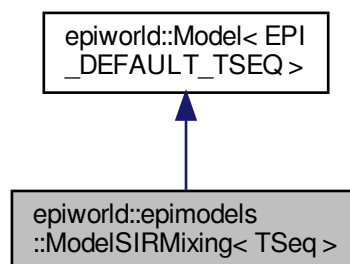
- `include/epiworld/models/sirlogit.hpp`

## 17.53 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

### 17.53.1 Constructor & Destructor Documentation

#### 17.53.1.1 `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 <a href="#">ModelSIRMixing</a> object.
<i>vname</i>	The name of the <a href="#">ModelSIRMixing</a> 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

17.53.1.2 [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 <a href="#">ModelSIRMixing</a> 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.

## 17.53.2 Member Function Documentation

### 17.53.2.1 `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 >`.

### 17.53.2.2 `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"> <li>The proportion of non-infected individuals who have recovered.</li> </ul>

Reimplemented from `epiworld::Model< EPI_DEFAULT_TSEQ >`.

### 17.53.2.3 `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

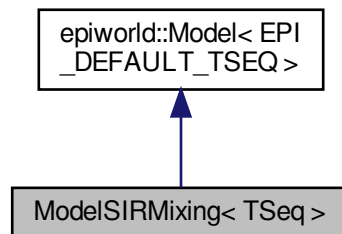
Reimplemented from `epiworld::Model< EPI_DEFAULT_TSEQ >`.

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

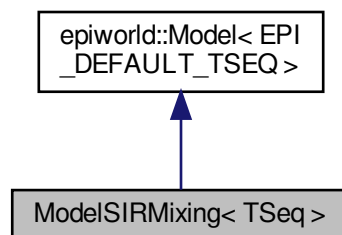
- `epiworld.hpp`

## 17.54 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

### 17.54.1 Constructor & Destructor Documentation

#### 17.54.1.1 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 <a href="#">ModelSIRMixing</a> object.
<i>vname</i>	The name of the <a href="#">ModelSIRMixing</a> 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

### 17.54.1.2 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 <a href="#">ModelSIRMixing</a> 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.

## 17.54.2 Member Function Documentation

### 17.54.2.1 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 >](#).

### 17.54.2.2 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"> <li>The proportion of non-infected individuals who have recovered.</li> </ul>

Reimplemented from [epiworld::Model< EPI\\_DEFAULT\\_TSEQ >](#).

#### 17.54.2.3 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:

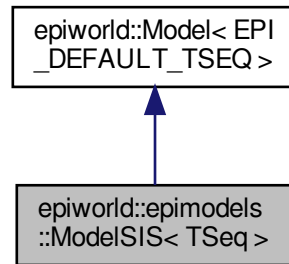
- include/epiworld/models/sirmixing.hpp

## 17.55 epiworld::epimodels::ModelSIS< TSeq > Class Template Reference

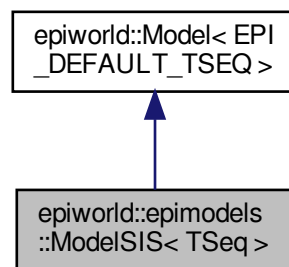
Template for a Susceptible-Infected-Susceptible (SIS) model.

```
#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

### 17.55.1 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:

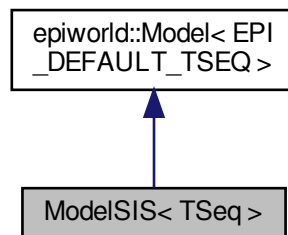
- epiworld.hpp

## 17.56 ModelSIS< TSeq > Class Template Reference

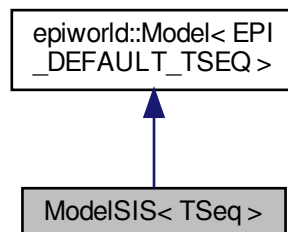
Template for a Susceptible-Infected-Susceptible (SIS) model.

```
#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

### 17.56.1 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:

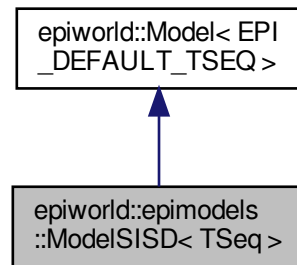
- `include/epiworld/models/sis.hpp`

## 17.57 `epiworld::epimodels::ModelSISD< TSeq >` Class Template Reference

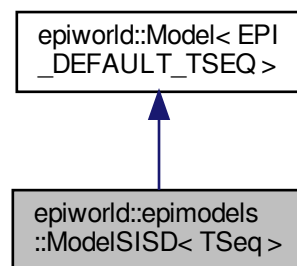
Template for a Susceptible-Infected-Susceptible-Deceased (SISD) model.

```
#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

### 17.57.1 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:

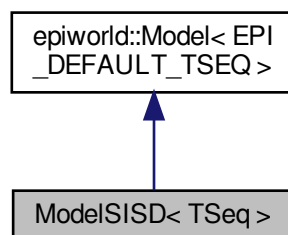
- epiworld.hpp

## 17.58 ModelSISD< TSeq > Class Template Reference

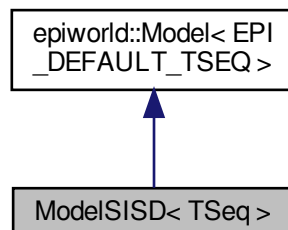
Template for a Susceptible-Infected-Susceptible-Deceased (SISD) model.

```
#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

### 17.58.1 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

## 17.59 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

### 17.59.1 Member Function Documentation

#### 17.59.1.1 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

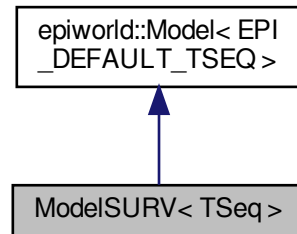
Reimplemented from [epiworld::Model< EPI\\_DEFAULT\\_TSEQ >](#).

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

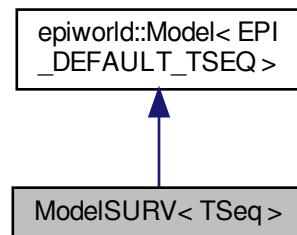
- epiworld.hpp

## 17.60 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****17.60.1 Member Function Documentation**

### 17.60.1.1 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`

## 17.61 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`

## 17.62 epiworld::PersonTools< TSeq > Class Template Reference

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

- `epiworld.hpp`

## 17.63 PersonTools< TSeq > Class Template Reference

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

- include/epiworld/config.hpp

## 17.64 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** ()

### 17.64.1 Detailed Description

A simple progress bar.

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

- epiworld.hpp

## 17.65 Progress Class Reference

A simple progress bar.

```
#include <progress.hpp>
```

### Public Member Functions

- **Progress** (int n\_, int width\_)
- void **start** ()
- void **next** ()
- void **end** ()

### 17.65.1 Detailed Description

A simple progress bar.

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

- include/epiworld/progress.hpp



## 17.66 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 >

#### 17.66.1 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

## 17.67 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 >

### 17.67.1 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

## 17.68 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

## 17.69 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

## 17.70 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

## 17.71 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)

**17.71.1 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

**17.72 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)

### 17.72.1 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

## 17.73 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

### 17.73.1 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

## 17.74 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

### 17.74.1 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

## 17.75 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 >

### 17.75.1 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

## 17.76 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 >

### 17.76.1 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



## 17.77 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 >

### 17.77.1 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

## 17.78 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 >

### 17.78.1 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

## 17.79 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 >

## 17.79.1 Detailed Description

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

Personalized data by the user.

#### Template Parameters

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

## 17.79.2 Constructor & Destructor Documentation

### 17.79.2.1 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

## 17.80 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 >

## 17.80.1 Detailed Description

```
template<typename TSeq>
class UserData< TSeq >
```

Personalized data by the user.

#### Template Parameters

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

## 17.80.2 Constructor & Destructor Documentation

### 17.80.2.1 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

## 17.81 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

#### 17.81.1 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

## 17.82 vecHasher< T > Struct Template Reference

Vector hasher.

```
#include <misc.hpp>
```

### Public Member Functions

- `std::size_t operator() (std::vector< T > const &dat) const` noexcept

#### 17.82.1 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

## 17.83 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)

## 17.83.1 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

## 17.84 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 ( <a href="#">Agent</a> ) 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)

### 17.84.1 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

## 17.85 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 >

### 17.85.1 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

## 17.86 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 >

### 17.86.1 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

## 17.87 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 >

### 17.87.1 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

## 17.88 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 >

### 17.88.1 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

## 17.89 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

## 17.90 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 18

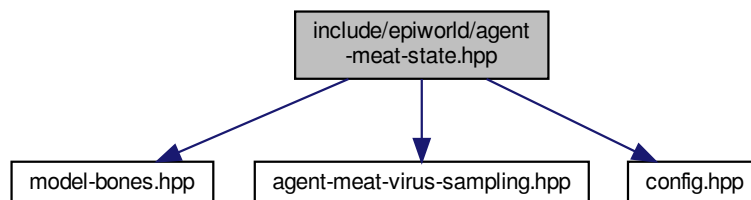
# File Documentation

### 18.1 include/epiworld/agent-meat-state.hpp File Reference

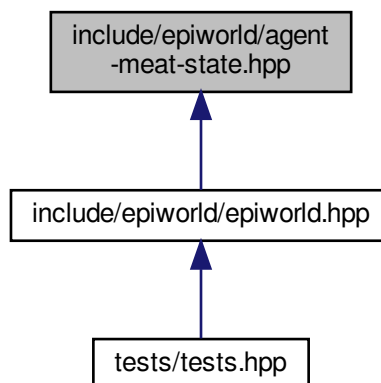
Sampling functions are getting big, so we keep them in a separate file.

```
#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)`

### 18.1.1 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

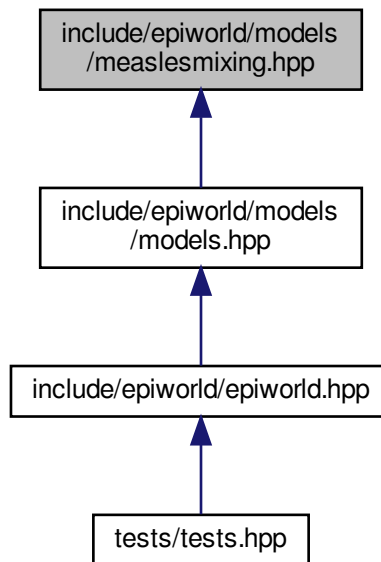
#### Copyright

Copyright (c) 2022

## 18.2 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)

### 18.2.1 Detailed Description

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

### 18.2.2 Macro Definition Documentation

### 18.2.2.1 GET\_MODEL

```
#define GET_MODEL(
 model,
 output)
```

**Value:**

```
auto * output = dynamic_cast< ModelMeaslesMixing<TSeq> * >((model)); \
assert((output) != nullptr);
```

### 18.2.2.2 SAMPLE\_FROM\_PROBS

```
#define SAMPLE_FROM_PROBS(
 n,
 ans)
```

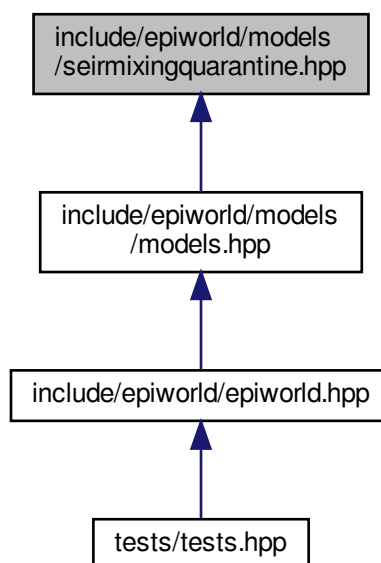
**Value:**

```
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]; \
}
```

## 18.3 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)`

### 18.3.1 Detailed Description

Template for a Susceptible-Exposed-Infected-Removed (SEIR) model with mixing, quarantine, and contact tracing.

### 18.3.2 Macro Definition Documentation

#### 18.3.2.1 GET\_MODEL

```
#define GET_MODEL(
 model,
 output)
```

##### Value:

```
auto * output = dynamic_cast< ModelSEIRMixingQuarantine<TSeq> * >((model)); \
assert((output) != nullptr);
```

#### 18.3.2.2 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 >, 99  
  Model< TSeq >, 115  
AdjList, 51  
  AdjList, 51  
  epiworld::AdjList, 53  
  read\_edgelist, 52  
Agent< TSeq >, 54  
  default\_rm\_entity, 58  
  operator(), 56  
  swap\_neighbors, 58  
AgentsSample  
  AgentsSample< TSeq >, 63  
  epiworld::AgentsSample< TSeq >, 65  
AgentsSample< TSeq >, 62  
  AgentsSample, 63  
  
clone\_ptr  
  epiworld::epimodels::ModelMeaslesMixing< TSeq  
    >, 132  
  epiworld::epimodels::ModelMeaslesSchool< TSeq  
    >, 146  
  epiworld::epimodels::ModelSEIRCONN< TSeq >,  
    159  
  epiworld::epimodels::ModelSEIRDCONN< TSeq  
    >, 172  
  epiworld::epimodels::ModelSEIRMixing< TSeq >,  
    179  
  epiworld::epimodels::ModelSEIRMixingQuarantine<  
    TSeq >, 189  
  epiworld::epimodels::ModelSIRCONN< TSeq >,  
    205  
  epiworld::epimodels::ModelSIRDCONN< TSeq >,  
    218  
  epiworld::epimodels::ModelSIRLogit< TSeq >,  
    224  
  epiworld::epimodels::ModelSIRMixing< TSeq >,  
    230  
  epiworld::Model< TSeq >, 99  
  Model< TSeq >, 115  
  ModelMeaslesMixing< TSeq >, 140  
  ModelMeaslesSchool< TSeq >, 151  
  ModelSEIRCONN< TSeq >, 162  
  ModelSEIRDCONN< TSeq >, 175  
  ModelSEIRMixing< TSeq >, 182  
  ModelSEIRMixingQuarantine< TSeq >, 197  
  ModelSIRCONN< TSeq >, 209  
  ModelSIRDCONN< TSeq >, 220  
  ModelSIRLogit< TSeq >, 227  
  ModelSIRMixing< TSeq >, 234  
  
DataBase< TSeq >, 65  
  get\_generation\_time, 68  
  get\_reproductive\_number, 68  
  get\_transition\_probability, 68  
  get\_transmissions, 69  
  operator==, 69, 70  
  record\_virus, 70  
default\_rm\_entity  
  Agent< TSeq >, 58  
  Entity< TSeq >, 80  
  epiworld::Agent< TSeq >, 62  
  epiworld::Entity< TSeq >, 81  
draw  
  epiworld::Model< TSeq >, 100  
  Model< TSeq >, 115  
  
Entities< TSeq >, 75  
Entities\_const< TSeq >, 77  
Entity  
  Entity< TSeq >, 79  
  epiworld::Entity< TSeq >, 81  
Entity< TSeq >, 79  
  default\_rm\_entity, 80  
  Entity, 79  
epiworld::AdjList, 52  
  AdjList, 53  
  read\_edgelist, 53  
epiworld::Agent< TSeq >, 59  
  default\_rm\_entity, 62  
  operator(), 61  
  swap\_neighbors, 61  
epiworld::AgentsSample< TSeq >, 64  
  AgentsSample, 65  
epiworld::DataBase< TSeq >, 71  
  get\_generation\_time, 73  
  get\_reproductive\_number, 73  
  get\_transition\_probability, 74  
  get\_transmissions, 74  
  operator==, 75  
  record\_virus, 75  
epiworld::Entities< TSeq >, 76  
epiworld::Entities\_const< TSeq >, 78  
epiworld::Entity< TSeq >, 80  
  default\_rm\_entity, 81  
  Entity, 81  
epiworld::epimodels::ModelDiffNet< TSeq >, 123  
epiworld::epimodels::ModelMeaslesMixing< TSeq >,  
  126  
  clone\_ptr, 132  
  get\_agent\_quarantine\_triggered, 132

- get\_contact\_matrix, 132
  - get\_isolation\_willingness, 133
  - get\_quarantine\_willingness, 133
  - initial\_states, 133
  - ModelMeaslesMixing, 129, 131
  - run, 134
  - set\_contact\_matrix, 134
- epiworld::epimodels::ModelMeaslesSchool< TSeq >, 143
  - clone\_ptr, 146
  - ModelMeaslesSchool, 145
  - quarantine\_agents, 147
  - reset, 147
- epiworld::epimodels::ModelSEIR< TSeq >, 152
  - initial\_states, 154
  - update\_exposed\_seir, 154
  - update\_infected\_seir, 154
- epiworld::epimodels::ModelSEIRCONN< TSeq >, 158
  - clone\_ptr, 159
  - initial\_states, 160
  - ModelSEIRCONN, 159
  - reset, 160
- epiworld::epimodels::ModelSEIRD< TSeq >, 164
  - ModelSEIRD, 165, 166
  - update\_exposed\_seir, 166
- epiworld::epimodels::ModelSEIRDCONN< TSeq >, 170
  - clone\_ptr, 172
  - initial\_states, 172
  - ModelSEIRDCONN, 171
  - reset, 172
- epiworld::epimodels::ModelSEIRMixing< TSeq >, 176
  - clone\_ptr, 179
  - initial\_states, 179
  - ModelSEIRMixing, 177, 178
  - reset, 179
- epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >, 184
  - clone\_ptr, 189
  - get\_agent\_quarantine\_triggered, 189
  - get\_contact\_matrix, 189
  - get\_isolation\_willingness, 190
  - get\_quarantine\_willingness, 190
  - initial\_states, 190
  - ModelSEIRMixingQuarantine, 186, 188
  - run, 191
  - set\_contact\_matrix, 191
- epiworld::epimodels::ModelSIR< TSeq >, 200
  - initial\_states, 201
- epiworld::epimodels::ModelSIRCONN< TSeq >, 204
  - clone\_ptr, 205
  - get\_n\_infected, 206
  - initial\_states, 206
  - ModelSIRCONN, 205
  - reset, 206
- epiworld::epimodels::ModelSIRD< TSeq >, 210
  - initial\_states, 213
  - ModelSIRD, 211
- epiworld::epimodels::ModelSIRDCONN< TSeq >, 216
  - clone\_ptr, 218
  - ModelSIRDCONN, 217
  - reset, 218
- epiworld::epimodels::ModelSIRLogit< TSeq >, 221
  - clone\_ptr, 224
  - ModelSIRLogit, 223
  - reset, 224
- epiworld::epimodels::ModelSIRMixing< TSeq >, 228
  - clone\_ptr, 230
  - initial\_states, 231
  - ModelSIRMixing, 229, 230
  - reset, 231
- epiworld::epimodels::ModelSIS< TSeq >, 235
- epiworld::epimodels::ModelSISD< TSeq >, 239
- epiworld::epimodels::ModelSURV< TSeq >, 242
  - reset, 244
- epiworld::Event< TSeq >, 82
  - Event, 83
- epiworld::GlobalEvent< TSeq >, 85
  - GlobalEvent, 86
- epiworld::LFMCMC< TData >, 88
- epiworld::Model< TSeq >, 91
  - add\_globalevent, 99
  - clone\_ptr, 99
  - draw, 100
  - events\_add, 100
  - events\_run, 101
  - initial\_states\_fun, 104
  - load\_agents\_entities\_ties, 101
  - rbinomd, 104
  - reset, 101
  - rexp, 104
  - rgammad, 104
  - rgeomd, 104
  - rlognormald, 105
  - rnbinomd, 105
  - rnormd, 105
  - rpoissd, 105
  - run\_multiple, 102
  - runifd, 105
  - set\_agents\_data, 102
  - set\_name, 103
  - time\_elapsed, 106
  - write\_data, 103
- epiworld::ModelDiagram, 123
- epiworld::PersonTools< TSeq >, 247
- epiworld::Progress, 248
- epiworld::Queue< TSeq >, 249
- epiworld::sampler, 43
  - make\_sample\_virus\_neighbors, 43
  - make\_update\_susceptible, 44
  - sample\_virus\_single, 44
- epiworld::SAMPLETYPE, 251
- epiworld::Tool< TSeq >, 251
- epiworld::ToolFunctions< TSeq >, 254
- epiworld::Tools< TSeq >, 255
- epiworld::Tools\_const< TSeq >, 257



- epiworld::UserData< TSeq >, 258
  - UserData, 260
- epiworld::vecHasher< T >, 262
- epiworld::Virus< TSeq >, 263
- epiworld::Viruses< TSeq >, 267
- epiworld::Viruses\_const< TSeq >, 269
- epiworld::VirusFunctions< TSeq >, 270
- Event
  - epiworld::Event< TSeq >, 83
  - Event< TSeq >, 85
- Event< TSeq >, 83
  - Event, 85
- events\_add
  - epiworld::Model< TSeq >, 100
  - Model< TSeq >, 116
- events\_run
  - epiworld::Model< TSeq >, 101
  - Model< TSeq >, 116
- get\_agent\_quarantine\_triggered
  - epiworld::epimodels::ModelMeaslesMixing< TSeq >, 132
  - epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >, 189
  - ModelMeaslesMixing< TSeq >, 141
  - ModelSEIRMixingQuarantine< TSeq >, 197
- get\_contact\_matrix
  - epiworld::epimodels::ModelMeaslesMixing< TSeq >, 132
  - epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >, 189
  - ModelMeaslesMixing< TSeq >, 141
  - ModelSEIRMixingQuarantine< TSeq >, 197
- get\_generation\_time
  - DataBase< TSeq >, 68
  - epiworld::DataBase< TSeq >, 73
- get\_isolation\_willingness
  - epiworld::epimodels::ModelMeaslesMixing< TSeq >, 133
  - epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >, 190
  - ModelMeaslesMixing< TSeq >, 141
  - ModelSEIRMixingQuarantine< TSeq >, 198
- GET\_MODEL
  - measlesmixing.hpp, 275
  - seirmixingquarantine.hpp, 277
- get\_n\_infected
  - epiworld::epimodels::ModelSIRCONN< TSeq >, 206
  - ModelSIRCONN< TSeq >, 209
- get\_quarantine\_willingness
  - epiworld::epimodels::ModelMeaslesMixing< TSeq >, 133
  - epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >, 190
  - ModelMeaslesMixing< TSeq >, 141
  - ModelSEIRMixingQuarantine< TSeq >, 198
- get\_reproductive\_number
  - DataBase< TSeq >, 68
- epiworld::DataBase< TSeq >, 73
- get\_transition\_probability
  - DataBase< TSeq >, 68
  - epiworld::DataBase< TSeq >, 74
- get\_transmissions
  - DataBase< TSeq >, 69
  - epiworld::DataBase< TSeq >, 74
- GlobalEvent
  - epiworld::GlobalEvent< TSeq >, 86
  - GlobalEvent< TSeq >, 87
- GlobalEvent< TSeq >, 87
  - GlobalEvent, 87
- include/epiworld/agent-meat-state.hpp, 273
- include/epiworld/models/measlesmixing.hpp, 275
- include/epiworld/models/seirmixingquarantine.hpp, 276
- initial\_states
  - epiworld::epimodels::ModelMeaslesMixing< TSeq >, 133
  - epiworld::epimodels::ModelSEIR< TSeq >, 154
  - epiworld::epimodels::ModelSEIRCONN< TSeq >, 160
  - epiworld::epimodels::ModelSEIRDCONN< TSeq >, 172
  - epiworld::epimodels::ModelSEIRMixing< TSeq >, 179
  - epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >, 190
  - epiworld::epimodels::ModelSIR< TSeq >, 201
  - epiworld::epimodels::ModelSIRCONN< TSeq >, 206
  - epiworld::epimodels::ModelSIRD< TSeq >, 213
  - epiworld::epimodels::ModelSIRMixing< TSeq >, 231
  - ModelMeaslesMixing< TSeq >, 142
  - ModelSEIR< TSeq >, 156
  - ModelSEIRCONN< TSeq >, 163
  - ModelSEIRDCONN< TSeq >, 175
  - ModelSEIRMixing< TSeq >, 183
  - ModelSEIRMixingQuarantine< TSeq >, 198
  - ModelSIR< TSeq >, 203
  - ModelSIRCONN< TSeq >, 209
  - ModelSIRD< TSeq >, 215
  - ModelSIRMixing< TSeq >, 234
- initial\_states\_fun
  - epiworld::Model< TSeq >, 104
  - Model< TSeq >, 120
- LFMCMC< TData >, 89
- load\_agents\_entities\_ties
  - epiworld::Model< TSeq >, 101
  - Model< TSeq >, 118
- make\_sample\_virus\_neighbors
  - epiworld::sampler, 43
  - sampler, 46
- make\_update\_susceptible
  - epiworld::sampler, 44
  - sampler, 47

- measlesmixing.hpp
  - GET\_MODEL, 275
  - SAMPLE\_FROM\_PROBS, 276
- Model< TSeq >, 106
  - add\_globlevent, 115
  - clone\_ptr, 115
  - draw, 115
  - events\_add, 116
  - events\_run, 116
  - initial\_states\_fun, 120
  - load\_agents\_entities\_ties, 118
  - rbinomd, 120
  - reset, 118
  - rexp, 120
  - rgammad, 121
  - rgeomd, 121
  - rlognormald, 121
  - rnbinomd, 121
  - rnormd, 121
  - rpoissd, 122
  - run\_multiple, 118
  - runifd, 122
  - set\_agents\_data, 119
  - set\_name, 119
  - time\_elapsed, 122
  - write\_data, 120
- ModelDiagram, 123
- ModelDiffNet< TSeq >, 125
- ModelMeaslesMixing
  - epiworld::epimodels::ModelMeaslesMixing< TSeq >, 129, 131
  - ModelMeaslesMixing< TSeq >, 138, 139
- ModelMeaslesMixing< TSeq >, 135
  - clone\_ptr, 140
  - get\_agent\_quarantine\_triggered, 141
  - get\_contact\_matrix, 141
  - get\_isolation\_willingness, 141
  - get\_quarantine\_willingness, 141
  - initial\_states, 142
  - ModelMeaslesMixing, 138, 139
  - run, 142
  - set\_contact\_matrix, 143
- ModelMeaslesSchool
  - epiworld::epimodels::ModelMeaslesSchool< TSeq >, 145
  - ModelMeaslesSchool< TSeq >, 150
- ModelMeaslesSchool< TSeq >, 148
  - clone\_ptr, 151
  - ModelMeaslesSchool, 150
  - quarantine\_agents, 151
  - reset, 151
- ModelSEIR< TSeq >, 155
  - initial\_states, 156
  - update\_exposed\_seir, 157
  - update\_infected\_seir, 157
- ModelSEIRCONN
  - epiworld::epimodels::ModelSEIRCONN< TSeq >, 159
  - ModelSEIRCONN< TSeq >, 162
  - clone\_ptr, 162
  - initial\_states, 163
  - ModelSEIRCONN, 162
  - reset, 163
- ModelSEIRD
  - epiworld::epimodels::ModelSEIRD< TSeq >, 165, 166
  - ModelSEIRD< TSeq >, 168, 169
- ModelSEIRD< TSeq >, 167
  - ModelSEIRD, 168, 169
  - update\_exposed\_seir, 169
- ModelSEIRDCONN
  - epiworld::epimodels::ModelSEIRDCONN< TSeq >, 171
  - ModelSEIRDCONN< TSeq >, 174
- ModelSEIRDCONN< TSeq >, 173
  - clone\_ptr, 175
  - initial\_states, 175
  - ModelSEIRDCONN, 174
  - reset, 176
- ModelSEIRMixing
  - epiworld::epimodels::ModelSEIRMixing< TSeq >, 177, 178
  - ModelSEIRMixing< TSeq >, 181, 182
- ModelSEIRMixing< TSeq >, 180
  - clone\_ptr, 182
  - initial\_states, 183
  - ModelSEIRMixing, 181, 182
  - reset, 183
- ModelSEIRMixingQuarantine
  - epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >, 186, 188
  - ModelSEIRMixingQuarantine< TSeq >, 194, 196
- ModelSEIRMixingQuarantine< TSeq >, 192
  - clone\_ptr, 197
  - get\_agent\_quarantine\_triggered, 197
  - get\_contact\_matrix, 197
  - get\_isolation\_willingness, 198
  - get\_quarantine\_willingness, 198
  - initial\_states, 198
  - ModelSEIRMixingQuarantine, 194, 196
  - run, 199
  - set\_contact\_matrix, 199
- ModelSIR< TSeq >, 201
  - initial\_states, 203
- ModelSIRCONN
  - epiworld::epimodels::ModelSIRCONN< TSeq >, 205
  - ModelSIRCONN< TSeq >, 208
- ModelSIRCONN< TSeq >, 207
  - clone\_ptr, 209
  - get\_n\_infected, 209
  - initial\_states, 209
  - ModelSIRCONN, 208
  - reset, 210
- ModelSIRD

- epiworld::epimodels::ModelSIRD< TSeq >, 211
- ModelSIRD< TSeq >, 215
- ModelSIRD< TSeq >, 213
  - initial\_states, 215
  - ModelSIRD, 215
- ModelSIRDCONN
  - epiworld::epimodels::ModelSIRDCONN< TSeq >, 217
  - ModelSIRDCONN< TSeq >, 220
- ModelSIRDCONN< TSeq >, 219
  - clone\_ptr, 220
  - ModelSIRDCONN, 220
  - reset, 221
- ModelSIRLogit
  - epiworld::epimodels::ModelSIRLogit< TSeq >, 223
  - ModelSIRLogit< TSeq >, 226
- ModelSIRLogit< TSeq >, 225
  - clone\_ptr, 227
  - ModelSIRLogit, 226
  - reset, 227
- ModelSIRMixing
  - epiworld::epimodels::ModelSIRMixing< TSeq >, 229, 230
  - ModelSIRMixing< TSeq >, 233, 234
- ModelSIRMixing< TSeq >, 232
  - clone\_ptr, 234
  - initial\_states, 234
  - ModelSIRMixing, 233, 234
  - reset, 235
- ModelSIS< TSeq >, 238
- ModelSISD< TSeq >, 241
- ModelSURV< TSeq >, 245
  - reset, 246
- Network< Nettype, Nodetype, Edgetype >, 247
- operator()
  - Agent< TSeq >, 56
  - epiworld::Agent< TSeq >, 61
- operator==
  - DataBase< TSeq >, 69, 70
  - epiworld::DataBase< TSeq >, 75
- PersonTools< TSeq >, 248
- Progress, 248
- quarantine\_agents
  - epiworld::epimodels::ModelMeaslesSchool< TSeq >, 147
  - ModelMeaslesSchool< TSeq >, 151
- Queue< TSeq >, 249
- RandGraph, 250
- rbinomd
  - epiworld::Model< TSeq >, 104
  - Model< TSeq >, 120
- read\_edgelist
  - AdjList, 52
- epiworld::AdjList, 53
- record\_virus
  - DataBase< TSeq >, 70
  - epiworld::DataBase< TSeq >, 75
- reset
  - epiworld::epimodels::ModelMeaslesSchool< TSeq >, 147
  - epiworld::epimodels::ModelSEIRCONN< TSeq >, 160
  - epiworld::epimodels::ModelSEIRDCONN< TSeq >, 172
  - epiworld::epimodels::ModelSEIRMixing< TSeq >, 179
  - epiworld::epimodels::ModelSIRCONN< TSeq >, 206
  - epiworld::epimodels::ModelSIRDCONN< TSeq >, 218
  - epiworld::epimodels::ModelSIRLogit< TSeq >, 224
  - epiworld::epimodels::ModelSIRMixing< TSeq >, 231
  - epiworld::epimodels::ModelSURV< TSeq >, 244
  - epiworld::Model< TSeq >, 101
  - Model< TSeq >, 118
  - ModelMeaslesSchool< TSeq >, 151
  - ModelSEIRCONN< TSeq >, 163
  - ModelSEIRDCONN< TSeq >, 176
  - ModelSEIRMixing< TSeq >, 183
  - ModelSIRCONN< TSeq >, 210
  - ModelSIRDCONN< TSeq >, 221
  - ModelSIRLogit< TSeq >, 227
  - ModelSIRMixing< TSeq >, 235
  - ModelSURV< TSeq >, 246
- rexp
  - epiworld::Model< TSeq >, 104
  - Model< TSeq >, 120
- rgammad
  - epiworld::Model< TSeq >, 104
  - Model< TSeq >, 121
- rgeomd
  - epiworld::Model< TSeq >, 104
  - Model< TSeq >, 121
- rlognormald
  - epiworld::Model< TSeq >, 105
  - Model< TSeq >, 121
- rnbinomd
  - epiworld::Model< TSeq >, 105
  - Model< TSeq >, 121
- rnormd
  - epiworld::Model< TSeq >, 105
  - Model< TSeq >, 121
- rpoissd
  - epiworld::Model< TSeq >, 105
  - Model< TSeq >, 122
- run
  - epiworld::epimodels::ModelMeaslesMixing< TSeq >, 134

- epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >, 191
- ModelMeaslesMixing< TSeq >, 142
- ModelSEIRMixingQuarantine< TSeq >, 199
- run\_multiple
  - epiworld::Model< TSeq >, 102
  - Model< TSeq >, 118
- runifd
  - epiworld::Model< TSeq >, 105
  - Model< TSeq >, 122
- SAMPLE\_FROM\_PROBS
  - measlesmixing.hpp, 276
  - seirmixingquarantine.hpp, 277
- sample\_virus\_single
  - epiworld::sampler, 44
  - sampler, 47
- sampler, 46
  - make\_sample\_virus\_neighbors, 46
  - make\_update\_susceptible, 47
  - sample\_virus\_single, 47
- SAMPLETYPE, 251
- seirmixingquarantine.hpp
  - GET\_MODEL, 277
  - SAMPLE\_FROM\_PROBS, 277
- set\_agents\_data
  - epiworld::Model< TSeq >, 102
  - Model< TSeq >, 119
- set\_contact\_matrix
  - epiworld::epimodels::ModelMeaslesMixing< TSeq >, 134
  - epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >, 191
  - ModelMeaslesMixing< TSeq >, 143
  - ModelSEIRMixingQuarantine< TSeq >, 199
- set\_name
  - epiworld::Model< TSeq >, 103
  - Model< TSeq >, 119
- swap\_neighbors
  - Agent< TSeq >, 58
  - epiworld::Agent< TSeq >, 61
- time\_elapsed
  - epiworld::Model< TSeq >, 106
  - Model< TSeq >, 122
- Tool< TSeq >, 252
- ToolFunctions< TSeq >, 255
- Tools< TSeq >, 256
- Tools\_const< TSeq >, 257
- update\_exposed\_seir
  - epiworld::epimodels::ModelSEIR< TSeq >, 154
  - epiworld::epimodels::ModelSEIRD< TSeq >, 166
  - ModelSEIR< TSeq >, 157
  - ModelSEIRD< TSeq >, 169
- update\_infected\_seir
  - epiworld::epimodels::ModelSEIR< TSeq >, 154
  - ModelSEIR< TSeq >, 157
- UserData
  - epiworld::UserData< TSeq >, 260
  - UserData< TSeq >, 261
  - UserData< TSeq >, 260
  - UserData, 261
- vecHasher< T >, 262
- Virus< TSeq >, 265
- Viruses< TSeq >, 268
- Viruses\_const< TSeq >, 269
- VirusFunctions< TSeq >, 271
- write\_data
  - epiworld::Model< TSeq >, 103
  - Model< TSeq >, 120