

epiworld

0.8.2

Generated by Doxygen 1.9.1

1 Example: <tt>00-hello-world</tt>	1
2 Contributor Code of Conduct	3
3 ModelMeaslesMixing	5
3.1 Overview	5
3.2 Disease States	5
3.3 Key Features	6
3.3.1 Disease Progression	6
3.3.2 Population Mixing	6
3.3.3 Public Health Measures	6
3.4 Parameters	6
3.4.1 Disease Parameters	6
3.4.2 Detection and Isolation	7
3.4.3 Contact Tracing and Quarantine	7
3.4.4 Vaccination	7
3.5 Usage Example	7
3.6 Differences from Other Models	8
3.6.1 vs. ModelMeaslesSchool	8
3.6.2 vs. ModelSEIRMixingQuarantine	8
3.7 Testing	8
4 ModelMeaslesMixingRiskQuarantine	9
4.1 Overview	9
4.1.1 Key Features	9
4.2 Disease States	9
4.3 Risk Classification System	10
4.3.1 High Risk	10
4.3.2 Medium Risk	10
4.3.3 Low Risk	10
4.4 Enhanced Detection	11
4.5 Parameters	11
4.5.1 Disease Parameters	11
4.5.2 Detection and Isolation	11
4.5.3 Risk-based Quarantine	11
4.5.4 Contact Tracing	12
4.5.5 Vaccination	12
4.6 Usage Example	12
4.7 Differences from Other Models	12
4.7.1 vs. ModelMeaslesMixing	12
4.7.2 vs. ModelMeaslesSchool	13
4.8 Testing	13
4.9 Applications	13

4.10 Implementation Notes	13
5 MIT License	15
6 Mixing probabilities in connected model	17
6.1 Case 1: No grouping	17
6.2 Case 2: Grouping	18
7 Namespace Index	19
7.1 Namespace List	19
8 Hierarchical Index	21
8.1 Class Hierarchy	21
9 Class Index	25
9.1 Class List	25
10 File Index	29
10.1 File List	29
11 Namespace Documentation	31
11.1 epiworld::sampler Namespace Reference	31
11.1.1 Detailed Description	31
11.1.2 Function Documentation	31
11.1.2.1 make_sample_virus_neighbors()	31
11.1.2.2 make_update_susceptible()	32
11.1.2.3 sample_virus_single()	32
11.2 sampler Namespace Reference	34
11.2.1 Detailed Description	34
11.2.2 Function Documentation	34
11.2.2.1 make_sample_virus_neighbors()	34
11.2.2.2 make_update_susceptible()	35
11.2.2.3 sample_virus_single()	35
12 Class Documentation	39
12.1 AdjList Class Reference	39
12.1.1 Constructor & Destructor Documentation	39
12.1.1.1 AdjList()	39
12.1.2 Member Function Documentation	40
12.1.2.1 read_edgelist()	40
12.2 epiworld::AdjList Class Reference	40
12.2.1 Constructor & Destructor Documentation	41
12.2.1.1 AdjList()	41
12.2.2 Member Function Documentation	41
12.2.2.1 read_edgelist()	41

12.3 Agent< TSeq > Class Template Reference	42
12.3.1 Detailed Description	44
12.3.2 Member Function Documentation	44
12.3.2.1 operator()()	44
12.3.2.2 swap_neighbors()	46
12.3.3 Friends And Related Function Documentation	46
12.3.3.1 default_rm_entity	46
12.4 epiworld::Agent< TSeq > Class Template Reference	47
12.4.1 Detailed Description	49
12.4.2 Member Function Documentation	49
12.4.2.1 operator()()	49
12.4.2.2 swap_neighbors()	50
12.4.3 Friends And Related Function Documentation	50
12.4.3.1 default_rm_entity	50
12.5 AgentsSample< TSeq > Class Template Reference	50
12.5.1 Detailed Description	51
12.5.2 Constructor & Destructor Documentation	51
12.5.2.1 AgentsSample()	51
12.6 epiworld::AgentsSample< TSeq > Class Template Reference	52
12.6.1 Detailed Description	52
12.6.2 Constructor & Destructor Documentation	53
12.6.2.1 AgentsSample()	53
12.7 ContactTracing Class Reference	53
12.7.1 Detailed Description	54
12.7.2 Constructor & Destructor Documentation	54
12.7.2.1 ContactTracing()	54
12.7.3 Member Function Documentation	54
12.7.3.1 add_contact()	55
12.7.3.2 get_contact()	55
12.7.3.3 get_n_contacts()	55
12.7.3.4 print()	56
12.7.3.5 reset()	56
12.8 epiworld::ContactTracing Class Reference	56
12.8.1 Detailed Description	57
12.8.2 Constructor & Destructor Documentation	57
12.8.2.1 ContactTracing()	57
12.8.3 Member Function Documentation	57
12.8.3.1 add_contact()	58
12.8.3.2 get_contact()	58
12.8.3.3 get_n_contacts()	58
12.8.3.4 print()	59
12.8.3.5 reset()	59

12.9 DataBase< TSeq > Class Template Reference	59
12.9.1 Detailed Description	61
12.9.2 Member Function Documentation	62
12.9.2.1 get_generation_time()	62
12.9.2.2 get_reproductive_number()	62
12.9.2.3 get_transition_probability()	63
12.9.2.4 get_transmissions()	63
12.9.2.5 operator==() [1/3]	63
12.9.2.6 operator==() [2/3]	64
12.9.2.7 operator==() [3/3]	64
12.9.2.8 record_virus()	64
12.10 epiworld::DataBase< TSeq > Class Template Reference	65
12.10.1 Detailed Description	67
12.10.2 Member Function Documentation	67
12.10.2.1 get_generation_time()	67
12.10.2.2 get_reproductive_number()	67
12.10.2.3 get_transition_probability()	68
12.10.2.4 get_transmissions()	68
12.10.2.5 operator==()	69
12.10.2.6 record_virus()	69
12.11 Entities< TSeq > Class Template Reference	69
12.11.1 Detailed Description	70
12.12 epiworld::Entities< TSeq > Class Template Reference	70
12.12.1 Detailed Description	71
12.13 Entities_const< TSeq > Class Template Reference	71
12.13.1 Detailed Description	71
12.14 epiworld::Entities_const< TSeq > Class Template Reference	72
12.14.1 Detailed Description	72
12.15 Entity< TSeq > Class Template Reference	73
12.15.1 Constructor & Destructor Documentation	73
12.15.1.1 Entity()	73
12.15.2 Friends And Related Function Documentation	74
12.15.2.1 default_rm_entity	74
12.16 epiworld::Entity< TSeq > Class Template Reference	74
12.16.1 Constructor & Destructor Documentation	75
12.16.1.1 Entity()	75
12.16.2 Friends And Related Function Documentation	75
12.16.2.1 default_rm_entity	76
12.17 epiworld::Event< TSeq > Struct Template Reference	76
12.17.1 Detailed Description	76
12.17.2 Constructor & Destructor Documentation	77
12.17.2.1 Event()	77

12.18 Event< TSeq > Struct Template Reference	77
12.18.1 Detailed Description	78
12.18.2 Constructor & Destructor Documentation	79
12.18.2.1 Event()	79
12.19 epiworld::GlobalEvent< TSeq > Class Template Reference	79
12.19.1 Detailed Description	80
12.19.2 Constructor & Destructor Documentation	80
12.19.2.1 GlobalEvent()	80
12.20 GlobalEvent< TSeq > Class Template Reference	81
12.20.1 Detailed Description	81
12.20.2 Constructor & Destructor Documentation	81
12.20.2.1 GlobalEvent()	81
12.21 epiworld::LFMCMC< TData > Class Template Reference	82
12.21.1 Detailed Description	83
12.22 LFMCMC< TData > Class Template Reference	83
12.22.1 Detailed Description	84
12.23 epiworld::Model< TSeq > Class Template Reference	85
12.23.1 Detailed Description	93
12.23.2 Member Function Documentation	93
12.23.2.1 add_globalevent()	93
12.23.2.2 clone_ptr()	94
12.23.2.3 draw()	94
12.23.2.4 events_add()	94
12.23.2.5 events_run()	95
12.23.2.6 load_agents_entities_ties()	95
12.23.2.7 reset()	96
12.23.2.8 run_multiple()	96
12.23.2.9 set_agents_data()	96
12.23.2.10 set_name()	97
12.23.2.11 write_data()	97
12.23.3 Member Data Documentation	98
12.23.3.1 initial_states_fun	98
12.23.3.2 rbinomd	98
12.23.3.3 rexpd	98
12.23.3.4 rgammad	99
12.23.3.5 rgeomd	99
12.23.3.6 rlognormald	99
12.23.3.7 rnbinomd	99
12.23.3.8 rnormd	99
12.23.3.9 rpoissd	100
12.23.3.10 runifd	100
12.23.3.11 time_elapsed	100

12.24 Model< TSeq > Class Template Reference	100
12.24.1 Detailed Description	108
12.24.2 Member Function Documentation	109
12.24.2.1 add_globalevent()	109
12.24.2.2 clone_ptr()	109
12.24.2.3 draw()	109
12.24.2.4 events_add()	110
12.24.2.5 events_run()	110
12.24.2.6 load_agents_entities_ties()	112
12.24.2.7 reset()	112
12.24.2.8 run_multiple()	113
12.24.2.9 set_agents_data()	113
12.24.2.10 set_name()	113
12.24.2.11 write_data()	114
12.24.3 Member Data Documentation	114
12.24.3.1 initial_states_fun	114
12.24.3.2 rbinomd	114
12.24.3.3 rexpd	115
12.24.3.4 rgammad	115
12.24.3.5 rgeomd	115
12.24.3.6 rlognormald	115
12.24.3.7 rnbinomd	115
12.24.3.8 rnormd	116
12.24.3.9 rpoissd	116
12.24.3.10 runifd	116
12.24.3.11 time_elapsed	116
12.25 epiworld::ModelDiagram Class Reference	117
12.26 ModelDiagram Class Reference	117
12.27 epiworld::epimodels::ModelDiffNet< TSeq > Class Template Reference	117
12.27.1 Detailed Description	118
12.28 ModelDiffNet< TSeq > Class Template Reference	119
12.28.1 Detailed Description	120
12.29 epiworld::epimodels::ModelMeaslesMixing< TSeq > Class Template Reference	120
12.29.1 Detailed Description	122
12.29.2 Constructor & Destructor Documentation	123
12.29.2.1 ModelMeaslesMixing() [1/2]	123
12.29.2.2 ModelMeaslesMixing() [2/2]	125
12.29.3 Member Function Documentation	126
12.29.3.1 clone_ptr()	126
12.29.3.2 get_agent_quarantine_triggered()	126
12.29.3.3 get_contact_matrix()	127
12.29.3.4 get_isolation_willingness()	127

12.29.3.5 get_quarantine_willingness()	127
12.29.3.6 initial_states()	127
12.29.3.7 run()	128
12.29.3.8 set_contact_matrix()	128
12.30 ModelMeaslesMixing< TSeq > Class Template Reference	129
12.30.1 Detailed Description	131
12.30.2 Constructor & Destructor Documentation	132
12.30.2.1 ModelMeaslesMixing() [1/2]	132
12.30.2.2 ModelMeaslesMixing() [2/2]	133
12.30.3 Member Function Documentation	134
12.30.3.1 clone_ptr()	134
12.30.3.2 get_agent_quarantine_triggered()	135
12.30.3.3 get_contact_matrix()	135
12.30.3.4 get_isolation_willingness()	135
12.30.3.5 get_quarantine_willingness()	136
12.30.3.6 initial_states()	136
12.30.3.7 run()	136
12.30.3.8 set_contact_matrix()	137
12.31 epiworld::epimodels::ModelMeaslesMixingRiskQuarantine< TSeq > Class Template Reference	137
12.31.1 Detailed Description	139
12.31.2 Constructor & Destructor Documentation	140
12.31.2.1 ModelMeaslesMixingRiskQuarantine() [1/2]	140
12.31.2.2 ModelMeaslesMixingRiskQuarantine() [2/2]	141
12.31.3 Member Function Documentation	142
12.31.3.1 clone_ptr()	142
12.31.3.2 get_contact_matrix()	142
12.31.3.3 get_days_quarantine_triggered()	143
12.31.3.4 get_isolation_willingness()	143
12.31.3.5 get_quarantine_risk_level()	143
12.31.3.6 get_quarantine_willingness()	143
12.31.3.7 initial_states()	144
12.31.3.8 run()	144
12.31.3.9 set_contact_matrix()	144
12.32 ModelMeaslesMixingRiskQuarantine< TSeq > Class Template Reference	145
12.32.1 Detailed Description	147
12.32.2 Constructor & Destructor Documentation	147
12.32.2.1 ModelMeaslesMixingRiskQuarantine() [1/2]	147
12.32.2.2 ModelMeaslesMixingRiskQuarantine() [2/2]	149
12.32.3 Member Function Documentation	150
12.32.3.1 clone_ptr()	150
12.32.3.2 get_contact_matrix()	150
12.32.3.3 get_days_quarantine_triggered()	150

12.32.3.4 get_isolation_willingness()	151
12.32.3.5 get_quarantine_risk_level()	151
12.32.3.6 get_quarantine_willingness()	151
12.32.3.7 initial_states()	151
12.32.3.8 run()	152
12.32.3.9 set_contact_matrix()	152
12.33 epiworld::epimodels::ModelMeaslesSchool< TSeq > Class Template Reference	153
12.33.1 Detailed Description	154
12.33.2 Constructor & Destructor Documentation	155
12.33.2.1 ModelMeaslesSchool()	155
12.33.3 Member Function Documentation	156
12.33.3.1 clone_ptr()	156
12.33.3.2 quarantine_agents()	156
12.33.3.3 reset()	157
12.34 ModelMeaslesSchool< TSeq > Class Template Reference	157
12.34.1 Detailed Description	159
12.34.2 Constructor & Destructor Documentation	159
12.34.2.1 ModelMeaslesSchool()	159
12.34.3 Member Function Documentation	160
12.34.3.1 clone_ptr()	160
12.34.3.2 quarantine_agents()	161
12.34.3.3 reset()	161
12.35 epiworld::epimodels::ModelSEIR< TSeq > Class Template Reference	162
12.35.1 Detailed Description	163
12.35.2 Member Function Documentation	163
12.35.2.1 initial_states()	163
12.35.3 Member Data Documentation	164
12.35.3.1 update_exposed_seir	164
12.35.3.2 update_infected_seir	164
12.36 ModelSEIR< TSeq > Class Template Reference	165
12.36.1 Detailed Description	166
12.36.2 Member Function Documentation	166
12.36.2.1 initial_states()	166
12.36.3 Member Data Documentation	167
12.36.3.1 update_exposed_seir	167
12.36.3.2 update_infected_seir	167
12.37 epiworld::epimodels::ModelSEIRCONN< TSeq > Class Template Reference	167
12.37.1 Constructor & Destructor Documentation	168
12.37.1.1 ModelSEIRCONN()	169
12.37.2 Member Function Documentation	169
12.37.2.1 clone_ptr()	169
12.37.2.2 initial_states()	169

12.37.2.3 reset()	170
12.38 ModelSEIRCONN< TSeq > Class Template Reference	170
12.38.1 Constructor & Destructor Documentation	171
12.38.1.1 ModelSEIRCONN()	172
12.38.2 Member Function Documentation	172
12.38.2.1 clone_ptr()	172
12.38.2.2 initial_states()	172
12.38.2.3 reset()	173
12.39 epiworld::epimodels::ModelSEIRD< TSeq > Class Template Reference	173
12.39.1 Detailed Description	175
12.39.2 Constructor & Destructor Documentation	175
12.39.2.1 ModelSEIRD() [1/2]	175
12.39.2.2 ModelSEIRD() [2/2]	176
12.39.3 Member Data Documentation	176
12.39.3.1 update_exposed_seir	176
12.40 ModelSEIRD< TSeq > Class Template Reference	177
12.40.1 Detailed Description	178
12.40.2 Constructor & Destructor Documentation	178
12.40.2.1 ModelSEIRD() [1/2]	178
12.40.2.2 ModelSEIRD() [2/2]	179
12.40.3 Member Data Documentation	179
12.40.3.1 update_exposed_seir	179
12.41 epiworld::epimodels::ModelSEIRDCONN< TSeq > Class Template Reference	180
12.41.1 Constructor & Destructor Documentation	181
12.41.1.1 ModelSEIRDCONN()	181
12.41.2 Member Function Documentation	182
12.41.2.1 clone_ptr()	182
12.41.2.2 initial_states()	182
12.41.2.3 reset()	182
12.42 ModelSEIRDCONN< TSeq > Class Template Reference	183
12.42.1 Constructor & Destructor Documentation	184
12.42.1.1 ModelSEIRDCONN()	184
12.42.2 Member Function Documentation	185
12.42.2.1 clone_ptr()	185
12.42.2.2 initial_states()	185
12.42.2.3 reset()	186
12.43 epiworld::epimodels::ModelSEIRMixing< TSeq > Class Template Reference	186
12.43.1 Constructor & Destructor Documentation	187
12.43.1.1 ModelSEIRMixing() [1/2]	188
12.43.1.2 ModelSEIRMixing() [2/2]	188
12.43.2 Member Function Documentation	189
12.43.2.1 clone_ptr()	189

12.43.2.2 initial_states()	189
12.43.2.3 reset()	190
12.44 ModelSEIRMixing< TSeq > Class Template Reference	190
12.44.1 Constructor & Destructor Documentation	191
12.44.1.1 ModelSEIRMixing() [1/2]	191
12.44.1.2 ModelSEIRMixing() [2/2]	192
12.44.2 Member Function Documentation	192
12.44.2.1 clone_ptr()	193
12.44.2.2 initial_states()	193
12.44.2.3 reset()	193
12.45 epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq > Class Template Reference	194
12.45.1 Detailed Description	196
12.45.2 Constructor & Destructor Documentation	196
12.45.2.1 ModelSEIRMixingQuarantine() [1/2]	197
12.45.2.2 ModelSEIRMixingQuarantine() [2/2]	198
12.45.3 Member Function Documentation	199
12.45.3.1 clone_ptr()	199
12.45.3.2 get_agent_quarantine_triggered()	199
12.45.3.3 get_contact_matrix()	200
12.45.3.4 get_isolation_willingness()	200
12.45.3.5 get_quarantine_willingness()	200
12.45.3.6 initial_states()	200
12.45.3.7 run()	201
12.45.3.8 set_contact_matrix()	201
12.46 ModelSEIRMixingQuarantine< TSeq > Class Template Reference	202
12.46.1 Detailed Description	204
12.46.2 Constructor & Destructor Documentation	204
12.46.2.1 ModelSEIRMixingQuarantine() [1/2]	205
12.46.2.2 ModelSEIRMixingQuarantine() [2/2]	206
12.46.3 Member Function Documentation	207
12.46.3.1 clone_ptr()	207
12.46.3.2 get_agent_quarantine_triggered()	207
12.46.3.3 get_contact_matrix()	208
12.46.3.4 get_isolation_willingness()	208
12.46.3.5 get_quarantine_willingness()	208
12.46.3.6 initial_states()	208
12.46.3.7 run()	209
12.46.3.8 set_contact_matrix()	209
12.47 epiworld::epimodels::ModelSIR< TSeq > Class Template Reference	210
12.47.1 Detailed Description	211
12.47.2 Member Function Documentation	211
12.47.2.1 initial_states()	211

12.48 ModelSIR< TSeq > Class Template Reference	211
12.48.1 Detailed Description	212
12.48.2 Member Function Documentation	213
12.48.2.1 initial_states()	213
12.49 epiworld::epimodels::ModelSIRCONN< TSeq > Class Template Reference	214
12.49.1 Constructor & Destructor Documentation	215
12.49.1.1 ModelSIRCONN()	215
12.49.2 Member Function Documentation	215
12.49.2.1 clone_ptr()	216
12.49.2.2 get_n_infected()	216
12.49.2.3 initial_states()	216
12.49.2.4 reset()	216
12.50 ModelSIRCONN< TSeq > Class Template Reference	217
12.50.1 Constructor & Destructor Documentation	218
12.50.1.1 ModelSIRCONN()	218
12.50.2 Member Function Documentation	219
12.50.2.1 clone_ptr()	219
12.50.2.2 get_n_infected()	219
12.50.2.3 initial_states()	219
12.50.2.4 reset()	220
12.51 epiworld::epimodels::ModelSIRD< TSeq > Class Template Reference	220
12.51.1 Detailed Description	221
12.51.2 Constructor & Destructor Documentation	221
12.51.2.1 ModelSIRD()	222
12.51.3 Member Function Documentation	223
12.51.3.1 initial_states()	223
12.52 ModelSIRD< TSeq > Class Template Reference	223
12.52.1 Detailed Description	224
12.52.2 Constructor & Destructor Documentation	225
12.52.2.1 ModelSIRD()	225
12.52.3 Member Function Documentation	225
12.52.3.1 initial_states()	225
12.53 epiworld::epimodels::ModelSIRDCCONN< TSeq > Class Template Reference	226
12.53.1 Constructor & Destructor Documentation	227
12.53.1.1 ModelSIRDCCONN()	227
12.53.2 Member Function Documentation	228
12.53.2.1 clone_ptr()	228
12.53.2.2 reset()	228
12.54 ModelSIRDCCONN< TSeq > Class Template Reference	229
12.54.1 Constructor & Destructor Documentation	230
12.54.1.1 ModelSIRDCCONN()	230
12.54.2 Member Function Documentation	230

12.54.2.1 <code>clone_ptr()</code>	230
12.54.2.2 <code>reset()</code>	231
12.55 <code>epiworld::epimodels::ModelSIRLogit< TSeq ></code> Class Template Reference	231
12.55.1 Detailed Description	232
12.55.2 Constructor & Destructor Documentation	233
12.55.2.1 <code>ModelSIRLogit()</code>	233
12.55.3 Member Function Documentation	234
12.55.3.1 <code>clone_ptr()</code>	234
12.55.3.2 <code>reset()</code>	234
12.56 <code>ModelSIRLogit< TSeq ></code> Class Template Reference	235
12.56.1 Detailed Description	236
12.56.2 Constructor & Destructor Documentation	236
12.56.2.1 <code>ModelSIRLogit()</code>	236
12.56.3 Member Function Documentation	237
12.56.3.1 <code>clone_ptr()</code>	237
12.56.3.2 <code>reset()</code>	237
12.57 <code>epiworld::epimodels::ModelSIRMixing< TSeq ></code> Class Template Reference	238
12.57.1 Constructor & Destructor Documentation	239
12.57.1.1 <code>ModelSIRMixing() [1/2]</code>	239
12.57.1.2 <code>ModelSIRMixing() [2/2]</code>	240
12.57.2 Member Function Documentation	240
12.57.2.1 <code>clone_ptr()</code>	241
12.57.2.2 <code>initial_states()</code>	241
12.57.2.3 <code>reset()</code>	241
12.58 <code>ModelSIRMixing< TSeq ></code> Class Template Reference	242
12.58.1 Constructor & Destructor Documentation	243
12.58.1.1 <code>ModelSIRMixing() [1/2]</code>	243
12.58.1.2 <code>ModelSIRMixing() [2/2]</code>	244
12.58.2 Member Function Documentation	244
12.58.2.1 <code>clone_ptr()</code>	244
12.58.2.2 <code>initial_states()</code>	244
12.58.2.3 <code>reset()</code>	245
12.59 <code>epiworld::epimodels::ModelSIS< TSeq ></code> Class Template Reference	245
12.59.1 Detailed Description	247
12.60 <code>ModelSIS< TSeq ></code> Class Template Reference	248
12.60.1 Detailed Description	249
12.61 <code>epiworld::epimodels::ModelSISD< TSeq ></code> Class Template Reference	249
12.61.1 Detailed Description	250
12.62 <code>ModelSISD< TSeq ></code> Class Template Reference	251
12.62.1 Detailed Description	252
12.63 <code>epiworld::epimodels::ModelSURV< TSeq ></code> Class Template Reference	252
12.63.1 Member Function Documentation	254

12.63.1.1 reset()	254
12.64 ModelSURV< TSeq > Class Template Reference	255
12.64.1 Member Function Documentation	256
12.64.1.1 reset()	257
12.65 Network< Nettype, Nodetype, Edgetype > Class Template Reference	257
12.66 epiworld::PersonTools< TSeq > Class Template Reference	257
12.67 PersonTools< TSeq > Class Template Reference	258
12.68 epiworld::Progress Class Reference	258
12.68.1 Detailed Description	258
12.69 Progress Class Reference	258
12.69.1 Detailed Description	258
12.70 epiworld::Queue< TSeq > Class Template Reference	259
12.70.1 Detailed Description	259
12.71 Queue< TSeq > Class Template Reference	259
12.71.1 Detailed Description	260
12.72 RandGraph Class Reference	260
12.73 epiworld::SAMPLETYPE Class Reference	261
12.74 SAMPLETYPE Class Reference	261
12.75 TestResults Struct Reference	261
12.76 epiworld::Tool< TSeq > Class Template Reference	261
12.76.1 Detailed Description	263
12.77 Tool< TSeq > Class Template Reference	263
12.77.1 Detailed Description	264
12.78 epiworld::ToolFunctions< TSeq > Class Template Reference	264
12.78.1 Detailed Description	265
12.79 ToolFunctions< TSeq > Class Template Reference	265
12.79.1 Detailed Description	265
12.80 epiworld::Tools< TSeq > Class Template Reference	266
12.80.1 Detailed Description	266
12.81 Tools< TSeq > Class Template Reference	266
12.81.1 Detailed Description	267
12.82 epiworld::Tools_const< TSeq > Class Template Reference	267
12.82.1 Detailed Description	268
12.83 Tools_const< TSeq > Class Template Reference	268
12.83.1 Detailed Description	268
12.84 epiworld::UserData< TSeq > Class Template Reference	269
12.84.1 Detailed Description	270
12.84.2 Constructor & Destructor Documentation	270
12.84.2.1 UserData()	270
12.85 UserData< TSeq > Class Template Reference	270
12.85.1 Detailed Description	271
12.85.2 Constructor & Destructor Documentation	272

12.85.2.1 UserData()	272
12.86 epiworld::vecHasher< T > Struct Template Reference	272
12.86.1 Detailed Description	272
12.87 vecHasher< T > Struct Template Reference	273
12.87.1 Detailed Description	273
12.88 epiworld::Virus< TSeq > Class Template Reference	273
12.88.1 Detailed Description	275
12.89 Virus< TSeq > Class Template Reference	275
12.89.1 Detailed Description	277
12.90 epiworld::Viruses< TSeq > Class Template Reference	278
12.90.1 Detailed Description	278
12.91 Viruses< TSeq > Class Template Reference	278
12.91.1 Detailed Description	279
12.92 epiworld::Viruses_const< TSeq > Class Template Reference	279
12.92.1 Detailed Description	280
12.93 Viruses_const< TSeq > Class Template Reference	280
12.93.1 Detailed Description	280
12.94 epiworld::VirusFunctions< TSeq > Class Template Reference	281
12.95 VirusFunctions< TSeq > Class Template Reference	281
13 File Documentation	283
13.1 include/epiworld/agent-meat-state.hpp File Reference	283
13.1.1 Detailed Description	284
13.2 include/epiworld/models/measlesmixing.hpp File Reference	285
13.2.1 Detailed Description	285
13.2.2 Macro Definition Documentation	285
13.2.2.1 GET_MODEL	286
13.2.2.2 SAMPLE_FROM_PROBS	286
13.3 include/epiworld/models/measlesmixingriskquarantine.hpp File Reference	286
13.3.1 Detailed Description	287
13.3.2 Macro Definition Documentation	287
13.3.2.1 GET_MODEL	287
13.3.2.2 SAMPLE_FROM_PROBS	287
13.4 include/epiworld/models/seirmixingquarantine.hpp File Reference	288
13.4.1 Detailed Description	288
13.4.2 Macro Definition Documentation	288
13.4.2.1 GET_MODEL	289
13.4.2.2 SAMPLE_FROM_PROBS	289
Index	291

Chapter 1

Example: <tt>00-hello-world</tt>

Output from the program:

Chapter 2

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 3

ModelMeaslesMixing

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

3.1 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

3.2 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

3.3 Key Features

3.3.1 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

3.3.2 Population Mixing

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

3.3.3 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

3.4 Parameters

3.4.1 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)

3.4.2 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

3.4.3 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

3.4.4 Vaccination

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

3.5 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();
```

3.6 Differences from Other Models

3.6.1 vs. ModelMeaslesSchool

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

3.6.2 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

3.7 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 4

ModelMeaslesMixingRiskQuarantine

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

4.1 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.

4.1.1 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

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** - Individuals who have recovered and gained immunity

4.3 Risk Classification System

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

4.3.1 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)

4.3.2 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)

4.3.3 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.

4.4 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

4.5 Parameters

4.5.1 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

4.5.2 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

4.5.3 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

4.5.4 Contact Tracing

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

4.5.5 Vaccination

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

4.6 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();
```

4.7 Differences from Other Models

4.7.1 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

4.7.2 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

4.8 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.

4.9 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

4.10 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 5

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 6

Mixing probabilities in connected model

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

6.1 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.

6.2 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 7

Namespace Index

7.1 Namespace List

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

epiworld::sampler	Functions for sampling viruses	31
sampler	Functions for sampling viruses	34

Chapter 8

Hierarchical Index

8.1 Class Hierarchy

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

AdjList	39
epiworld::AdjList	40
Agent< TSeq >	42
epiworld::Agent< TSeq >	47
Agent< EPI_DEFAULT_TSEQ >	42
AgentsSample< TSeq >	50
epiworld::AgentsSample< TSeq >	52
ContactTracing	53
epiworld::ContactTracing	56
DataBase< TSeq >	59
epiworld::DataBase< TSeq >	65
Entities< TSeq >	69
epiworld::Entities< TSeq >	70
Entities_const< TSeq >	71
epiworld::Entities_const< TSeq >	72
Entity< TSeq >	73
epiworld::Entity< TSeq >	74
Entity< EPI_DEFAULT_TSEQ >	73
epiworld::Event< TSeq >	76
Event< TSeq >	77
epiworld::GlobalEvent< TSeq >	79
GlobalEvent< TSeq >	81
epiworld::LFMCMC< TData >	82
LFMCMC< TData >	83
epiworld::Model< TSeq >	85
Model< TSeq >	100
ModelMeaslesMixingRiskQuarantine< TSeq >	145
ModelMeaslesSchool< TSeq >	157
epiworld::Model< EPI_DEFAULT_TSEQ >	85
ModelDiffNet< TSeq >	119
ModelMeaslesMixing< TSeq >	129
ModelSEIR< TSeq >	165
ModelSEIRCONN< TSeq >	170
ModelSEIRD< TSeq >	177
ModelSEIRDCCONN< TSeq >	183

ModelSEIRMixing< TSeq >	190
ModelSEIRMixingQuarantine< TSeq >	202
ModelSIR< TSeq >	211
ModelSIRCONN< TSeq >	217
ModelSIRD< TSeq >	223
ModelSIRDCCONN< TSeq >	229
ModelSIRLogit< TSeq >	235
ModelSIRMixing< TSeq >	242
ModelSIS< TSeq >	248
ModelSISD< TSeq >	251
ModelSURV< TSeq >	255
epiworld::epimodels::ModelDiffNet< TSeq >	117
epiworld::epimodels::ModelMeaslesMixing< TSeq >	120
epiworld::epimodels::ModelMeaslesMixingRiskQuarantine< TSeq >	137
epiworld::epimodels::ModelMeaslesSchool< TSeq >	153
epiworld::epimodels::ModelSEIR< TSeq >	162
epiworld::epimodels::ModelSEIRCONN< TSeq >	167
epiworld::epimodels::ModelSEIRD< TSeq >	173
epiworld::epimodels::ModelSEIRDCCONN< TSeq >	180
epiworld::epimodels::ModelSEIRMixing< TSeq >	186
epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >	194
epiworld::epimodels::ModelSIR< TSeq >	210
epiworld::epimodels::ModelSIRCONN< TSeq >	214
epiworld::epimodels::ModelSIRD< TSeq >	220
epiworld::epimodels::ModelSIRDCCONN< TSeq >	226
epiworld::epimodels::ModelSIRLogit< TSeq >	231
epiworld::epimodels::ModelSIRMixing< TSeq >	238
epiworld::epimodels::ModelSIS< TSeq >	245
epiworld::epimodels::ModelSISD< TSeq >	249
epiworld::epimodels::ModelSURV< TSeq >	252
epiworld::ModelDiagram	117
ModelDiagram	117
Network< Nettype, Nodetype, Edgetype >	257
epiworld::PersonTools< TSeq >	257
PersonTools< TSeq >	258
epiworld::Progress	258
Progress	258
epiworld::Queue< TSeq >	259
Queue< TSeq >	259
RandGraph	260
epiworld::SAMPLETYPE	261
SAMPLETYPE	261
TestResults	261
epiworld::Tool< TSeq >	261
Tool< TSeq >	263
epiworld::ToolFunctions< TSeq >	264
ToolFunctions< TSeq >	265
epiworld::Tools< TSeq >	266
Tools< TSeq >	266
epiworld::Tools_const< TSeq >	267
Tools_const< TSeq >	268
epiworld::UserData< TSeq >	269
UserData< TSeq >	270
epiworld::vecHasher< T >	272
vecHasher< T >	273
epiworld::Virus< TSeq >	273
Virus< TSeq >	275
epiworld::Viruses< TSeq >	278

Viruses< TSeq >	278
epiworld::Viruses_const< TSeq >	279
Viruses_const< TSeq >	280
epiworld::VirusFunctions< TSeq >	281
VirusFunctions< TSeq >	281

Chapter 9

Class Index

9.1 Class List

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

AdjList	39
epiworld::AdjList	40
Agent< TSeq >	
Agent (agents)	42
epiworld::Agent< TSeq >	
Agent (agents)	47
AgentsSample< TSeq >	
Sample of agents	50
epiworld::AgentsSample< TSeq >	
Sample of agents	52
ContactTracing	
Class for tracing contacts between agents	53
epiworld::ContactTracing	
Class for tracing contacts between agents	56
DataBase< TSeq >	
Statistical data about the process	59
epiworld::DataBase< TSeq >	
Statistical data about the process	65
Entities< TSeq >	
Set of Entities (useful for building iterators)	69
epiworld::Entities< TSeq >	
Set of Entities (useful for building iterators)	70
Entities_const< TSeq >	
Set of Entities (const) (useful for iterators)	71
epiworld::Entities_const< TSeq >	
Set of Entities (const) (useful for iterators)	72
Entity< TSeq >	
Entity data for update an agent	73
epiworld::Entity< TSeq >	
Entity data for update an agent	74
epiworld::Event< TSeq >	
Event data for update an agent	76
Event< TSeq >	
Event data for update an agent	77
epiworld::GlobalEvent< TSeq >	
Template for a Global Event	79
GlobalEvent< TSeq >	
Template for a Global Event	81

epiworld::LFMCMC< TData >	
Likelihood-Free Markov Chain Monte Carlo	82
LFMCMC< TData >	
Likelihood-Free Markov Chain Monte Carlo	83
epiworld::Model< TSeq >	
Core class of epiworld	85
Model< TSeq >	
Core class of epiworld	100
epiworld::ModelDiagram	117
ModelDiagram	117
epiworld::epimodels::ModelDiffNet< TSeq >	
Template for a Network Diffusion Model	117
ModelDiffNet< TSeq >	
Template for a Network Diffusion Model	119
epiworld::epimodels::ModelMeaslesMixing< TSeq >	
Measles model with population mixing, quarantine, and contact tracing	120
ModelMeaslesMixing< TSeq >	
Measles model with population mixing, quarantine, and contact tracing	129
epiworld::epimodels::ModelMeaslesMixingRiskQuarantine< TSeq >	
Measles model with population mixing and risk-based quarantine strategies	137
ModelMeaslesMixingRiskQuarantine< TSeq >	
Measles model with population mixing and risk-based quarantine strategies	145
epiworld::epimodels::ModelMeaslesSchool< TSeq >	
Template for a Measles model with quarantine	153
ModelMeaslesSchool< TSeq >	
Template for a Measles model with quarantine	157
epiworld::epimodels::ModelSEIR< TSeq >	
Template for a Susceptible-Exposed-Infected-Removed (SEIR) model	162
ModelSEIR< TSeq >	
Template for a Susceptible-Exposed-Infected-Removed (SEIR) model	165
epiworld::epimodels::ModelSEIRCONN< TSeq >	167
ModelSEIRCONN< TSeq >	170
epiworld::epimodels::ModelSEIRD< TSeq >	
Template for a Susceptible-Exposed-Infected-Removed-Deceased (SEIRD) model	173
ModelSEIRD< TSeq >	
Template for a Susceptible-Exposed-Infected-Removed-Deceased (SEIRD) model	177
epiworld::epimodels::ModelSEIRDCCONN< TSeq >	180
ModelSEIRDCCONN< TSeq >	183
epiworld::epimodels::ModelSEIRMixing< TSeq >	186
ModelSEIRMixing< TSeq >	190
epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >	
SEIR model with mixing, quarantine, and contact tracing	194
ModelSEIRMixingQuarantine< TSeq >	
SEIR model with mixing, quarantine, and contact tracing	202
epiworld::epimodels::ModelSIR< TSeq >	
Template for a Susceptible-Infected-Removed (SIR) model	210
ModelSIR< TSeq >	
Template for a Susceptible-Infected-Removed (SIR) model	211
epiworld::epimodels::ModelSIRCONN< TSeq >	214
ModelSIRCONN< TSeq >	217
epiworld::epimodels::ModelSIRD< TSeq >	
Template for a Susceptible-Infected-Removed-Deceased (SIRD) model	220
ModelSIRD< TSeq >	
Template for a Susceptible-Infected-Removed-Deceased (SIRD) model	223
epiworld::epimodels::ModelSIRDCCONN< TSeq >	226
ModelSIRDCCONN< TSeq >	229
epiworld::epimodels::ModelSIRLogit< TSeq >	
Template for a Susceptible-Infected-Removed (SIR) model	231

ModelSIRLogit< TSeq >	235
Template for a Susceptible-Infected-Removed (SIR) model	235
epiworld::epimodels::ModelSIRMixing< TSeq >	238
ModelSIRMixing< TSeq >	242
epiworld::epimodels::ModelSIS< TSeq >	245
Template for a Susceptible-Infected-Susceptible (SIS) model	245
ModelSIS< TSeq >	248
Template for a Susceptible-Infected-Susceptible (SIS) model	248
epiworld::epimodels::ModelSISD< TSeq >	249
Template for a Susceptible-Infected-Susceptible-Deceased (SISD) model	249
ModelSISD< TSeq >	251
Template for a Susceptible-Infected-Susceptible-Deceased (SISD) model	251
epiworld::epimodels::ModelSURV< TSeq >	252
ModelSURV< TSeq >	255
Network< Nettype, Nodetype, Edgetype >	257
epiworld::PersonTools< TSeq >	257
PersonTools< TSeq >	258
epiworld::Progress	258
A simple progress bar	258
Progress	258
A simple progress bar	258
epiworld::Queue< TSeq >	259
Controls which agents are verified at each step	259
Queue< TSeq >	259
Controls which agents are verified at each step	259
RandGraph	260
epiworld::SAMPLETYPE	261
SAMPLETYPE	261
TestResults	261
epiworld::Tool< TSeq >	261
Tools for defending the agent against the virus	261
Tool< TSeq >	263
Tools for defending the agent against the virus	263
epiworld::ToolFunctions< TSeq >	264
Helper class to store the functions avoiding multiple shared_pointers (we have only one for the four of these)	264
ToolFunctions< TSeq >	265
Helper class to store the functions avoiding multiple shared_pointers (we have only one for the four of these)	265
epiworld::Tools< TSeq >	266
Set of tools (useful for building iterators)	266
Tools< TSeq >	266
Set of tools (useful for building iterators)	266
epiworld::Tools_const< TSeq >	267
Set of Tools (const) (useful for iterators)	267
Tools_const< TSeq >	268
Set of Tools (const) (useful for iterators)	268
epiworld::UserData< TSeq >	269
Personalized data by the user	269
UserData< TSeq >	270
Personalized data by the user	270
epiworld::vecHasher< T >	272
Vector hasher	272
vecHasher< T >	273
Vector hasher	273
epiworld::Virus< TSeq >	273
Virus	273

Virus< TSeq >	
Virus
epiworld::Viruses< TSeq >	
Set of viruses (useful for building iterators)
Viruses< TSeq >	
Set of viruses (useful for building iterators)
epiworld::Viruses_const< TSeq >	
Set of Viruses (const) (useful for iterators)
Viruses_const< TSeq >	
Set of Viruses (const) (useful for iterators)
epiworld::VirusFunctions< TSeq >
VirusFunctions< TSeq >

Chapter 10

File Index

10.1 File List

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

epiworld.hpp	??
include/epiworld/ adjlist-bones.hpp	??
include/epiworld/ adjlist-meat.hpp	??
include/epiworld/ agent-bones.hpp	??
include/epiworld/ agent-events-meat.hpp	??
include/epiworld/ agent-meat-state.hpp	
Sampling functions are getting big, so we keep them in a separate file	283
include/epiworld/ agent-meat-virus-sampling.hpp	??
include/epiworld/ agent-meat.hpp	??
include/epiworld/ agentssample-bones.hpp	??
include/epiworld/ config.hpp	??
include/epiworld/ contacttracing-bones.hpp	??
include/epiworld/ database-bones.hpp	??
include/epiworld/ database-meat.hpp	??
include/epiworld/ entities-bones.hpp	??
include/epiworld/ entity-bones.hpp	??
include/epiworld/ entity-distribute-meat.hpp	??
include/epiworld/ entity-meat.hpp	??
include/epiworld/ epiworld-macros.hpp	??
include/epiworld/ epiworld.hpp	??
include/epiworld/ globalevent-bones.hpp	??
include/epiworld/ globalevent-meat.hpp	??
include/epiworld/ misc.hpp	??
include/epiworld/ model-bones.hpp	??
include/epiworld/ model-meat-print.hpp	??
include/epiworld/ model-meat.hpp	??
include/epiworld/ modeldiagram-bones.hpp	??
include/epiworld/ modeldiagram-meat.hpp	??
include/epiworld/ network-bones.hpp	??
include/epiworld/ progress.hpp	??
include/epiworld/ queue-bones.hpp	??
include/epiworld/ randgraph.hpp	??
include/epiworld/ random_graph.hpp	??
include/epiworld/ seq_processing.hpp	??
include/epiworld/ tool-bones.hpp	??

include/epiworld/ tool-distribute-meat.hpp	??
include/epiworld/ tool-meat.hpp	??
include/epiworld/ tools-bones.hpp	??
include/epiworld/ userdata-bones.hpp	??
include/epiworld/ userdata-meat.hpp	??
include/epiworld/ virus-bones.hpp	??
include/epiworld/ virus-distribute-meat.hpp	??
include/epiworld/ virus-meat.hpp	??
include/epiworld/ viruses-bones.hpp	??
include/epiworld/math/ distributions.hpp	??
include/epiworld/math/ lfmcmc.hpp	??
include/epiworld/math/lfmcmc/ lfmcmc-bones.hpp	??
include/epiworld/math/lfmcmc/ lfmcmc-meat-print.hpp	??
include/epiworld/math/lfmcmc/ lfmcmc-meat.hpp	??
include/epiworld/models/ diffnet.hpp	??
include/epiworld/models/ globalevents.hpp	??
include/epiworld/models/ init-functions.hpp	??
include/epiworld/models/ measlesmixing.hpp	
Template for a Measles model with population mixing, quarantine, and contact tracing	285
include/epiworld/models/measlesmixingriskquarantine.hpp	
Template for a Measles model with population mixing and risk-based quarantine	286
include/epiworld/models/ measlesschool.hpp	??
include/epiworld/models/ models.hpp	??
include/epiworld/models/ seir.hpp	??
include/epiworld/models/ seirconnected.hpp	??
include/epiworld/models/ seird.hpp	??
include/epiworld/models/ seirdconnected.hpp	??
include/epiworld/models/ seirmixing.hpp	??
include/epiworld/models/seirmixingquarantine.hpp	
Template for a Susceptible-Exposed-Infected-Removed (SEIR) model with mixing, quarantine, and contact tracing	288
include/epiworld/models/ sir.hpp	??
include/epiworld/models/ sirconnected.hpp	??
include/epiworld/models/ sird.hpp	??
include/epiworld/models/ sirdconnected.hpp	??
include/epiworld/models/ sirlogit.hpp	??
include/epiworld/models/ sirmixing.hpp	??
include/epiworld/models/ sis.hpp	??
include/epiworld/models/ sisd.hpp	??
include/epiworld/models/ surveillance.hpp	??
tests/ tests.hpp	??

Chapter 11

Namespace Documentation

11.1 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 > *(< a href="#">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)

11.1.1 Detailed Description

Functions for sampling viruses.

11.1.2 Function Documentation

11.1.2.1 [make_sample_virus_neighbors\(\)](#)

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
std::function< Virus< TSeq > *(< a href="#">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;

11.1.2.2 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;

11.1.2.3 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;

11.2 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)

11.2.1 Detailed Description

Functions for sampling viruses.

11.2.2 Function Documentation

11.2.2.1 [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;

11.2.2.2 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;

11.2.2.3 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 12

Class Documentation

12.1 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.

12.1.1 Constructor & Destructor Documentation

12.1.1.1 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

12.1.2 Member Function Documentation**12.1.2.1 `read_edgelist()`**

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

Read an edgelist.

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

Parameters

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

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

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

12.2 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.

12.2.1 Constructor & Destructor Documentation

12.2.1.1 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

12.2.2 Member Function Documentation

12.2.2.1 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

12.3 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	<i>Tool to add</i>
virus	<i>Virus to add</i>
state_new	<i>state after the change</i>
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	<i>A pointer to a virus.</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)

- 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)

12.3.1 Detailed Description

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

Agent (agents)

Template Parameters

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

12.3.2 Member Function Documentation

12.3.2.1 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&

12.3.2.2 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>	

12.3.3 Friends And Related Function Documentation**12.3.3.1 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

12.4 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	Tool to add
virus	Virus to add
state_new	state after the change
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)`

12.4.1 Detailed Description

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

`Agent` (agents)

Template Parameters

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

12.4.2 Member Function Documentation

12.4.2.1 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

<code>j</code>	
----------------	--

Returns

`double&`

12.4.2.2 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>	

12.4.3 Friends And Related Function Documentation

12.4.3.1 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

12.5 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`

12.5.1 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

<code>TSeq</code>	
-------------------	--

12.5.2 Constructor & Destructor Documentation

12.5.2.1 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> \leftarrow	
<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

12.6 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](#)

12.6.1 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>	
-------------	--

12.6.2 Constructor & Destructor Documentation**12.6.2.1 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

12.7 ContactTracing Class Reference

Class for tracing contacts between agents.

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

Public Member Functions

- `ContactTracing` (`size_t n_agents, size_t max_contacts`)

Construct a new Contact Tracing object.
- `void add_contact` (`size_t agent_a, size_t agent_b, size_t day`)

Add a contact between two agents at a given day.
- `size_t get_n_contacts` (`size_t agent`)

Get the number of contacts for an agent.
- `std::pair<size_t, size_t> get_contact` (`size_t agent, size_t idx`)

Get the contact object.
- `void reset` (`size_t n_agents, size_t max_contacts`)

Reset the contact tracing data.
- `void print` (`size_t agent`)

Print the contacts of an agent.

12.7.1 Detailed Description

Class for tracing contacts between agents.

The class assumes that contacts are stored in a matrix-like structure, where rows are agents and columns are contacts in a column-major format. Each entry stores the id of the contact and the day of the contact is stored in a parallel array.

The maximum number of contacts per agent is fixed at initialization time. If more contacts are added, they will overwrite previous contacts in a circular buffer fashion.

12.7.2 Constructor & Destructor Documentation

12.7.2.1 ContactTracing()

```
ContactTracing::ContactTracing (
    size_t n_agents,
    size_t max_contacts ) [inline]
```

Construct a new Contact Tracing object.

Parameters

<code>n_agents</code>	Agents in the system, usually <code>Model<TSeq>::size()</code> .
<code>max_contacts</code>	Maximum number of contacts to track per agent.

12.7.3 Member Function Documentation

12.7.3.1 add_contact()

```
void ContactTracing::add_contact (
    size_t agent_a,
    size_t agent_b,
    size_t day ) [inline]
```

Add a contact between two agents at a given day.

Parameters

<i>agent_a</i>	Agent id (usually infectious agent)
<i>agent_b</i>	Agent id (usually susceptible agent)
<i>day</i>	Day of the contact (usually Model<TSeq>::today()).

12.7.3.2 get_contact()

```
std::pair< size_t, size_t > ContactTracing::get_contact (
    size_t agent,
    size_t idx ) [inline]
```

Get the contact object.

Parameters

<i>agent</i>	Source agent (id)
<i>idx</i>	Index of the contact (0 to get_n_contacts(agent)-1)

Returns

`std::pair<size_t, size_t>` with (`contact_id, contact_day`)

Exceptions

<code>std::out_of_range</code>	if <code>idx</code> is out of range.
--------------------------------	--------------------------------------

12.7.3.3 get_n_contacts()

```
size_t ContactTracing::get_n_contacts (
    size_t agent ) [inline]
```

Get the number of contacts for an agent.

Parameters

<i>agent</i>	Agent id
--------------	--------------------------

Returns

`size_t` Number of contacts recorded for that agent (can be more than `max_contacts`)

12.7.3.4 `print()`

```
void ContactTracing::print (
    size_t agent ) [inline]
```

Print the contacts of an agent.

Parameters

<i>agent</i>	Agent id
--------------	--------------------------

12.7.3.5 `reset()`

```
void ContactTracing::reset (
    size_t n_agents,
    size_t max_contacts ) [inline]
```

Reset the contact tracing data.

Usually called by `Model<TSeq>::reset()`.

Parameters

<i>n_agents</i>	Number of agents
<i>max_contacts</i>	Maximum number of contacts to track per agent

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

- include/epiworld/contacttracing-bones.hpp

12.8 epiworld::ContactTracing Class Reference

Class for tracing contacts between agents.

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

Public Member Functions

- `ContactTracing (size_t n_agents, size_t max_contacts)`
`Construct a new Contact Tracing object.`
- `void add_contact (size_t agent_a, size_t agent_b, size_t day)`
`Add a contact between two agents at a given day.`
- `size_t get_n_contacts (size_t agent)`
`Get the number of contacts for an agent.`
- `std::pair< size_t, size_t > get_contact (size_t agent, size_t idx)`
`Get the contact object.`
- `void reset (size_t n_agents, size_t max_contacts)`
`Reset the contact tracing data.`
- `void print (size_t agent)`
`Print the contacts of an agent.`

12.8.1 Detailed Description

Class for tracing contacts between agents.

The class assumes that contacts are stored in a matrix-like structure, where rows are agents and columns are contacts in a column-major format. Each entry stores the id of the contact and the day of the contact is stored in a parallel array.

The maximum number of contacts per agent is fixed at initialization time. If more contacts are added, they will overwrite previous contacts in a circular buffer fashion.

12.8.2 Constructor & Destructor Documentation

12.8.2.1 ContactTracing()

```
ContactTracing::ContactTracing (
    size_t n_agents,
    size_t max_contacts ) [inline]
```

Construct a new Contact Tracing object.

Parameters

<code>n_agents</code>	Agents in the system, usually <code>Model<TSeq>::size()</code> .
<code>max_contacts</code>	Maximum number of contacts to track per agent.

12.8.3 Member Function Documentation

12.8.3.1 add_contact()

```
void ContactTracing::add_contact (
    size_t agent_a,
    size_t agent_b,
    size_t day ) [inline]
```

Add a contact between two agents at a given day.

Parameters

<i>agent_a</i>	Agent id (usually infectious agent)
<i>agent_b</i>	Agent id (usually susceptible agent)
<i>day</i>	Day of the contact (usually Model<TSeq>::today()).

12.8.3.2 get_contact()

```
std::pair< size_t, size_t > ContactTracing::get_contact (
    size_t agent,
    size_t idx ) [inline]
```

Get the contact object.

Parameters

<i>agent</i>	Source agent (id)
<i>idx</i>	Index of the contact (0 to <code>get_n_contacts(agent)-1</code>)

Returns

`std::pair<size_t, size_t>` with (`contact_id, contact_day`)

Exceptions

<code>std::out_of_range</code>	if <code>idx</code> is out of range.
--------------------------------	--------------------------------------

12.8.3.3 get_n_contacts()

```
size_t ContactTracing::get_n_contacts (
    size_t agent ) [inline]
```

Get the number of contacts for an agent.

Parameters

<i>agent</i>	Agent id
--------------	--------------------------

Returns

`size_t` Number of contacts recorded for that agent (can be more than `max_contacts`)

12.8.3.4 `print()`

```
void ContactTracing::print (
    size_t agent ) [inline]
```

Print the contacts of an agent.

Parameters

<i>agent</i>	Agent id
--------------	--------------------------

12.8.3.5 `reset()`

```
void ContactTracing::reset (
    size_t n_agents,
    size_t max_contacts ) [inline]
```

Reset the contact tracing data.

Usually called by [Model<TSeq>::reset\(\)](#).

Parameters

<i>n_agents</i>	Number of agents
<i>max_contacts</i>	Maximum number of contacts to track per agent

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

- epiworld.hpp

12.9 DataBase< TSeq > Class Template Reference

Statistical data about the process.

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

Public Member Functions

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

Get recorded information from the model

Parameters

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

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)

12.9.1 Detailed Description

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

Statistical data about the process.

Template Parameters

TSeq

12.9.2 Member Function Documentation

12.9.2.1 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.

12.9.2.2 get_reproductive_number()

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

Computes the reproductive number of each case.

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

Parameters

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

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

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

12.9.2.3 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 >
```

12.9.2.4 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>	

12.9.2.5 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,

12.9.2.6 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,
```

12.9.2.7 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
```

12.9.2.8 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

12.10 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

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

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)

12.10.1 Detailed Description

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

Statistical data about the process.

Template Parameters

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

12.10.2 Member Function Documentation

12.10.2.1 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.

12.10.2.2 get_reproductive_number()

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

Computes the reproductive number of each case.

By definition, whereas it computes R₀ (basic reproductive number) or 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

12.10.2.3 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 >
```

12.10.2.4 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>	

12.10.2.5 operator==()

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

< Date of the transmission eve

< Id of the sour

< Id of the targ

< Id of the varia

< Date when the source acquired the varia
```

12.10.2.6 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

12.11 Entities< TSeq > Class Template Reference

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

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

Public Member Functions

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

Friends

- class `Entity< TSeq >`
- class `Agent< TSeq >`

12.11.1 Detailed Description

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

Set of `Entities` (useful for building iterators)

Template Parameters

<code>TSeq</code>	
-------------------	--

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

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

12.12 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 >

12.12.1 Detailed Description

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

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

Template Parameters

TSeq	
------	--

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

- epiworld.hpp

12.13 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 >

12.13.1 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

12.14 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 >

12.14.1 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

12.15 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)

12.15.1 Constructor & Destructor Documentation

12.15.1.1 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

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

12.15.2 Friends And Related Function Documentation

12.15.2.1 `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

12.16 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)

12.16.1 Constructor & Destructor Documentation

12.16.1.1 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.

12.16.2 Friends And Related Function Documentation

12.16.2.1 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

12.17 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**

12.17.1 Detailed Description

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

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

Template Parameters

TSeq	
------	--

12.17.2 Constructor & Destructor Documentation**12.17.2.1 Event()**

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

Construct a new [Event](#) object.

All the parameters are rather optional.

Parameters

<i>agent_</i>	Agent over who the action will happen
<i>virus_</i>	Virus to add
<i>tool_</i>	Tool to add
<i>virus_idx</i>	Index of virus to be removed (if needed)
<i>tool_idx</i>	Index of tool to be removed (if needed)
<i>new_<- state_</i>	Next state
<i>queue_</i>	Efect 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:

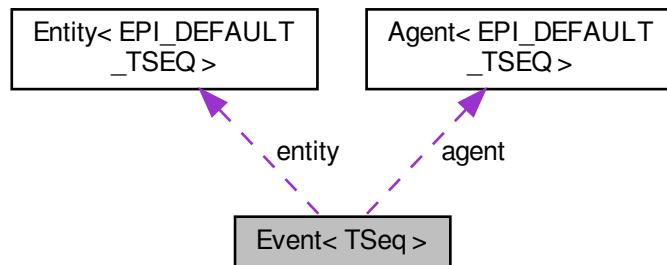
- epiworld.hpp

12.18 Event< TSeq > Struct Template Reference

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

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

Collaboration diagram for Event< TSeq >:



Public Member Functions

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

Construct a new `Event` object.

Public Attributes

- `Agent< TSeq > * agent`
- `VirusPtr< TSeq > virus`
- `ToolPtr< TSeq > tool`
- `Entity< TSeq > * entity`
- `epiworld_fast_int new_state`
- `epiworld_fast_int queue`
- `EventFun< TSeq > call`
- `int idx_agent`
- `int idx_object`

12.18.1 Detailed Description

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

`Event` data for update an agent.

Template Parameters

<code>TSeq</code>	
-------------------	--

12.18.2 Constructor & Destructor Documentation

12.18.2.1 Event()

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

Construct a new [Event](#) object.

All the parameters are rather optional.

Parameters

<i>agent_</i>	Agent over who the action will happen
<i>virus_</i>	Virus to add
<i>tool_</i>	Tool to add
<i>virus_idx</i>	Index of virus to be removed (if needed)
<i>tool_idx</i>	Index of tool to be removed (if needed)
<i>new_<- state_</i>	Next state
<i>queue_</i>	Efect 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

12.19 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

12.19.1 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.

12.19.2 Constructor & Destructor Documentation

12.19.2.1 GlobalEvent()

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

Construct a new Global Event object.

Parameters

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

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

- epiworld.hpp

12.20 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

12.20.1 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.

12.20.2 Constructor & Destructor Documentation

12.20.2.1 GlobalEvent()

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

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

Parameters

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

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

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

12.21 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	<input type="button" value=""/>
-----	---------------------------------

- void **set_rand_engine** (std::shared_ptr< std::mt19937 > &eng)
- std::shared_ptr< std::mt19937 > & **get_rand_endgine** ()
- 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)

12.21.1 Detailed Description

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

Likelihood-Free Markov Chain Monte Carlo.

Template Parameters

TData	Type of data that is generated
-------	--------------------------------

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

- epiworld.hpp

12.22 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_endgine ()`
- `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)`

12.22.1 Detailed Description

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

Likelihood-Free Markov Chain Monte Carlo.

Template Parameters

TData	Type of data that is generated
-------	--------------------------------

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

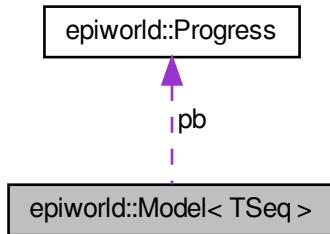
- include/epiworld/math/lfmcmc/lfmcmc-bones.hpp
- include/epiworld/math/lfmcmc/lfmcmc-meat-print.hpp
- include/epiworld/math/lfmcmc/lfmcmc-meat.hpp

12.23 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_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>
prevall	<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

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

<code>seed</code>	<i>Seed to be used for Pseudo-RNG.</i>
<code>ndays</code>	<i>Number of days (steps) of the simulation.</i>
<code>fun</code>	<i>In the case of run_multiple, 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 nexpperiments, 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

<code>proportion</code>	<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**

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

<code>lab</code>	<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

<code>proportions</code> ←	<i>Vector of proportions for each state.</i>
<code>_</code>	

<code>queue_</code>	<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 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 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 **rexpd**
- 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** ()

12.23.1 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.)
-------------	---

12.23.2 Member Function Documentation

12.23.2.1 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.

12.23.2.2 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>	<input type="checkbox"/>
-------------	--------------------------

Reimplemented in [ModelSIRMixing< TSeq >](#), [ModelSIRLogit< TSeq >](#), [ModelSIRDConn< TSeq >](#), [ModelSIRConn< TSeq >](#), [ModelSEIRMixingQuarantine< TSeq >](#), [ModelSEIRMixing< TSeq >](#), [ModelSEIRDConn< TSeq >](#), [ModelSEIRConn< TSeq >](#), [ModelMeaslesMixing< TSeq >](#), epiworld::epimodels::ModelMeaslesMixingRiskQuarantine< TSeq >, epiworld::epimodels::ModelMeaslesMixingQuarantine< TSeq >, epiworld::epimodels::ModelMeaslesSchool< TSeq >, epiworld::epimodels::ModelSIRMixing< TSeq >, epiworld::epimodels::ModelSEIRMixing< TSeq >, epiworld::epimodels::ModelSIRDConn< TSeq >, epiworld::epimodels::ModelSIRDConn< TSeq >, epiworld::epimodels::ModelSIRConn< TSeq >, and [epiworld::epimodels::ModelSIRConn< TSeq >](#).

12.23.2.3 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.

12.23.2.4 events_add()

```
template<typename TSeq >
void Model< TSeq >::events_add (
    Agent< TSeq > * agent_,
    VirusPtr< TSeq > virus_,
    ToolPtr< TSeq > tool_,
    Entity< TSeq > * entity_,
    epiworld_fast_int new_state_,
    epiworld_fast_int queue_,
    EventFun< TSeq > call_,
```

```
int idx_agent_,
int idx_object_) [inline], [protected]
```

Construct a new [Event](#) object.

Parameters

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

12.23.2.5 events_run()

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

Executes the stored action.

Parameters

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

12.23.2.6 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.

12.23.2.7 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::ModelMeaslesMixingRiskQuarantine< TSeq >](#), [epiworld::epimodels::ModelMeaslesMixing< TSeq >](#), [epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >](#), [epiworld::epimodels::ModelMeaslesSchool< TSeq >](#), [epiworld::epimodels::ModelSIRMixing< TSeq >](#), [epiworld::epimodels::ModelSIRLogit< TSeq >](#), [epiworld::epimodels::ModelSEIRDCONN< TSeq >](#), [epiworld::epimodels::ModelSIRCONN< TSeq >](#), and [epiworld::epimodels::ModelSEIRCONN< TSeq >](#)

12.23.2.8 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

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

12.23.2.9 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.

12.23.2.10 set_name()

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

Set the name object.

Parameters

<i>name</i>	<input type="text"/>
-------------	----------------------

12.23.2.11 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.

Parameters

<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

12.23.3 Member Data Documentation**12.23.3.1 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

12.23.3.2 rbinomd

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

Initial value:

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

12.23.3.3 rexpd

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

Initial value:

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

12.23.3.4 rgammad

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

Initial value:

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

12.23.3.5 rgeomd

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

Initial value:

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

12.23.3.6 rlognormald

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

Initial value:

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

12.23.3.7 rnbinomd

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

Initial value:

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

12.23.3.8 rnormd

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

Initial value:

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

12.23.3.9 rpoissd

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

Initial value:

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

12.23.3.10 runifd

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

Initial value:

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

12.23.3.11 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:

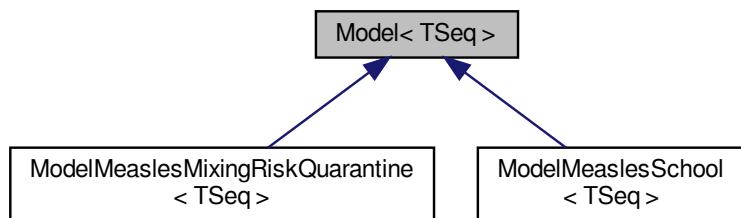
- epiworld.hpp

12.24 Model< TSeq > Class Template Reference

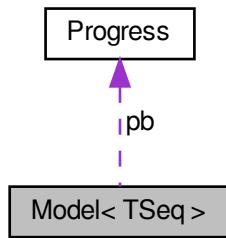
Core class of epiworld.

```
#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_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_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 **rbinom** ()
- int **rbinom** (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</i> to be added
t	<i>Tool</i> to be added
prevail	<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</i> <i>Filename of the edgelist file.</i>
skip	<i>int</i> <i>Number of lines to skip in fn.</i>
directed	<i>bool</i> <i>Whether the graph is directed or not.</i>
size	<i>Size of the network.</i>
al	<i>AdjList</i> <i>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 run_multiple, 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 nexpperiments, 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 > *)> fun, 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	<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 \leftarrow	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 **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 **rexpd**
- std::binomial_distribution **rbinomd**
- std::negative_binomial_distribution **rnbinnomd**
- std::geometric_distribution **rgeomd**
- std::poisson_distribution **rpoisssd**
- 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()**

12.24.1 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.)
-------------	---

12.24.2 Member Function Documentation**12.24.2.1 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.

12.24.2.2 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 >](#), and [ModelMeaslesMixingRiskQuarantine< TSeq >](#).

12.24.2.3 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.

12.24.2.4 events_add()

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

Construct a new [Event](#) object.

Parameters

<i>agent_</i>	Agent over which the action will be called
<i>virus_</i>	Virus pointer included in the action
<i>tool_</i>	Tool pointer included in the action
<i>entity_</i>	Entity pointer included in the action
<i>new_state_</i>	New state of the agent
<i>queue_</i>	Function the action will call
<i>idx_agent_</i>	Change in the queue
<i>idx_object_</i>	Location of agent in object.
<i>idx_object_</i>	Location of object in agent.

12.24.2.5 events_run()

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

Executes the stored action.

Parameters

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

12.24.2.6 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.

12.24.2.7 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 >, and [ModelMeaslesMixingRiskQuarantine](#)< TSeq >.

12.24.2.8 run_multiple()

```
template<typename TSeq >
void Model< TSeq >::run_multiple (
    epiworld_fast_uint ndays,
    epiworld_fast_uint nexpiments,
    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
--------------	---------------------------------

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

12.24.2.10 set_name()

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

Set the name object.

Parameters

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

12.24.2.11 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

12.24.3 Member Data Documentation

12.24.3.1 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

12.24.3.2 rbinomd

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

Initial value:

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

12.24.3.3 rexp_d

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

Initial value:

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

12.24.3.4 rgamma_d

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

Initial value:

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

12.24.3.5 rgeom_d

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

Initial value:

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

12.24.3.6 rlognormal_d

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

Initial value:

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

12.24.3.7 rnbinom_d

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

Initial value:

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

12.24.3.8 rnormd

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

Initial value:

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

12.24.3.9 rpoissd

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

Initial value:

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

12.24.3.10 runifd

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

Initial value:

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

12.24.3.11 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

12.25 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

12.26 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:

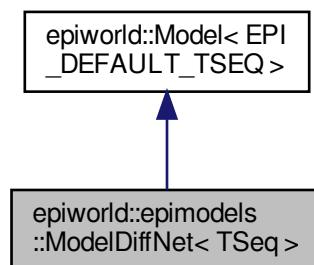
- include/epiworld/modeldiagram-bones.hpp
- include/epiworld/modeldiagram-meat.hpp

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

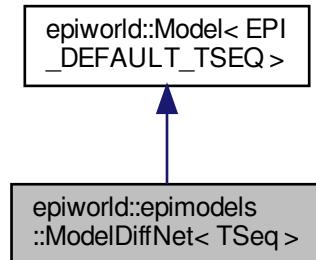
Template for a [Network Diffusion Model](#).

```
#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

12.27.1 Detailed Description

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

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

Parameters

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

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

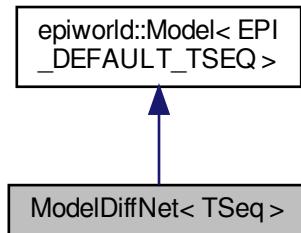
- epiworld.hpp

12.28 ModelDiffNet< TSeq > Class Template Reference

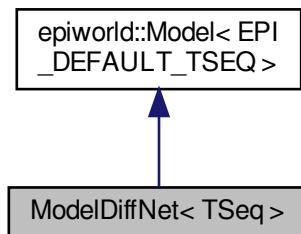
Template for a [Network Diffusion Model](#).

```
#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

12.28.1 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_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:

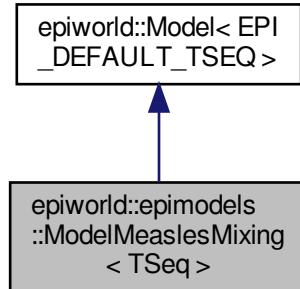
- include/epiworld/models/diffnet.hpp

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

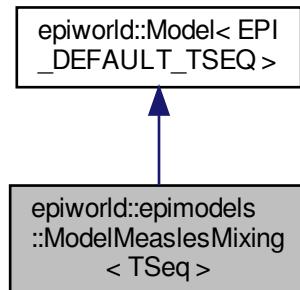
Measles model with population mixing, quarantine, and contact tracing.

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

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



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



Public Member Functions

- [`ModelMeaslesMixing \(ModelMeaslesMixing< TSeq > &model, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double vax_efficacy, epiworld_double vax_reduction_recovery_rate, epiworld_double incubation_period, epiworld_double prodromal_period, epiworld_double rash_period, std::vector< double > contact_matrix, epiworld_double hospitalization_rate, epiworld_double hospitalization_period, epiworld_double daysUndetected, epiworldFastInt quarantinePeriod, epiworld_double quarantineWillingness, epiworld_double isolationWillingness, epiworldFastInt isolationPeriod, epiworld_double propVaccinated, epiworld_double contactTracingSuccessRate=1.0, epiworldFastUint contactTracingDaysPrior=4u\)`](#)
Constructs a [ModelMeaslesMixing](#) object.
- [`ModelMeaslesMixing \(epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double vax_efficiency, epiworld_double vax_reduction_recovery_rate, epiworld_double incubation_period, epiworld_double prodromal_period, epiworld_double rash_period, std::vector< double > contact_matrix, epiworld_double hospitalization_rate, epiworld_double hospitalization_period, epiworld_double daysUndetected, epiworld_fast_int quarantinePeriod, epiworld_double quarantineWillingness, epiworld_double isolationWillingness, epiworld_fast_int isolationPeriod\)`](#)

```
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

12.29.1 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

TSeq	Type for genetic sequences (default: EPI_DEFAULT_TSEQ)
------	--

12.29.2 Constructor & Destructor Documentation

12.29.2.1 ModelMeaslesMixing() [1/2]

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

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

Constructs a [ModelMeaslesMixing](#) object.

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

Parameters

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

Parameters

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

12.29.2.2 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).

12.29.3 Member Function Documentation**12.29.3.1 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 >](#).

12.29.3.2 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

12.29.3.3 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

12.29.3.4 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

12.29.3.5 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

12.29.3.6 initial_states()

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

Set the initial states of the model.

Parameters

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

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

12.29.3.7 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 >](#).

12.29.3.8 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:

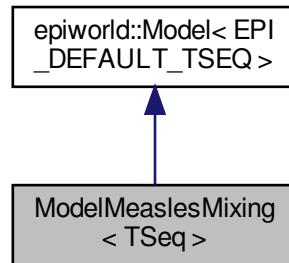
- [epiworld.hpp](#)

12.30 ModelMeaslesMixing< TSeq > Class Template Reference

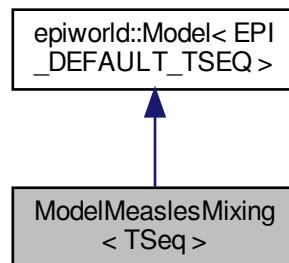
Measles model with population mixing, quarantine, and contact tracing.

```
#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_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

12.30.1 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)
-------------	--

12.30.2 Constructor & Destructor Documentation

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

Constructs a [ModelMeaslesMixing](#) object.

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

Parameters

<i>model</i>	A reference to an existing ModelMeaslesMixing object.
<i>n</i>	The number of entities in the model.
<i>prevalence</i>	The initial prevalence of the disease in the model.
<i>contact_rate</i>	The contact rate between entities in the model.
<i>transmission_rate</i>	The transmission rate of the disease in the model.
<i>vax_efficiency</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

12.30.2.2 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).

12.30.3 Member Function Documentation

12.30.3.1 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 >](#).

12.30.3.2 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

12.30.3.3 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

12.30.3.4 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

12.30.3.5 `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

12.30.3.6 `initial_states()`

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

Set the initial states of the model.

Parameters

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

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

12.30.3.7 `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 >](#).

12.30.3.8 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:

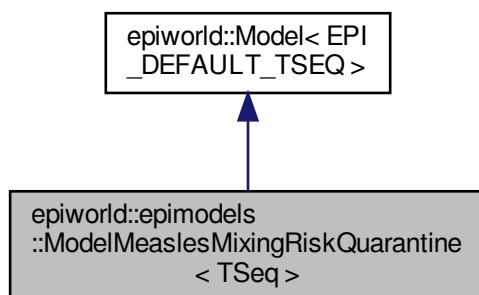
- [include/epiworld/models/measlesmixing.hpp](#)

12.31 epiworld::epimodels::ModelMeaslesMixingRiskQuarantine< TSeq > Class Template Reference

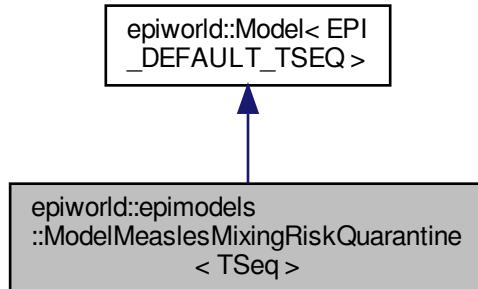
Measles model with population mixing and risk-based quarantine strategies.

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

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



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



Public Member Functions

- **ModelMeaslesMixingRiskQuarantine** (`ModelMeaslesMixingRiskQuarantine< TSeq > &model, epiworld<_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double vax_efficacy, 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 daysUndetected, epiworld_fast_int quarantinePeriod_high, epiworld_fast_int quarantinePeriod_medium, epiworld_fast_int quarantinePeriod_low, epiworld_double quarantineWillingness, epiworld_double isolationWillingness, epiworld_fast_int isolationPeriod, epiworld_double propVaccinated, epiworld_double detectionRateQuarantine, epiworld_double contactTracingSuccessRate=1.0, epiworld_fast_uint contactTracingDaysPrior=4u)`
Constructs a `ModelMeaslesMixingRiskQuarantine` object.
- **ModelMeaslesMixingRiskQuarantine** (`epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double vax_efficacy, 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 daysUndetected, epiworld_fast_int quarantinePeriod_high, epiworld_fast_int quarantinePeriod_medium, epiworld_fast_int quarantinePeriod_low, epiworld_double quarantineWillingness, epiworld_double isolationWillingness, epiworld_double propVaccinated, epiworld_double detectionRateQuarantine, epiworld_double contactTracingSuccessRate=1.0, epiworld_fast_uint contactTracingDaysPrior=4u)`
Constructs a `ModelMeaslesMixingRiskQuarantine` object.
- **ModelMeaslesMixingRiskQuarantine< 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.
- **ModelMeaslesMixingRiskQuarantine< 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.
- **auto get_contact_matrix () const**
Get the current contact matrix.

- auto `get_quarantine_willingness () const`
Get the quarantine willingness for all agents.
- auto `get_isolation_willingness () const`
Get the isolation willingness for all agents.
- const auto & `get_quarantine_risk_level () const`
Get the risk level assigned to each agent for quarantine purposes.
- auto `get_days_quarantine_triggered () const`
Get the total number of quarantines that have occurred.

Static Public Attributes

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

Additional Inherited Members

12.31.1 Detailed Description

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

Measles model with population mixing and risk-based quarantine strategies.

This class extends the Measles epidemiological model to support different quarantine strategies based on exposure risk levels:

- **High Risk:** Unvaccinated agents who share entity membership with the case
- **Medium Risk:** Unvaccinated agents who contacted an infected individual but don't share entity membership
- **Low Risk:** Other unvaccinated agents

Each risk level can have different quarantine durations, allowing for targeted public health interventions. The model also includes enhanced detection during active quarantine periods.

Disease progression follows the same states as [ModelMeaslesMixing](#): Susceptible → Exposed → Prodromal → Rash → Recovered

Template Parameters

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

12.31.2 Constructor & Destructor Documentation

12.31.2.1 ModelMeaslesMixingRiskQuarantine() [1/2]

```
template<typename TSeq >
ModelMeaslesMixingRiskQuarantine< TSeq >::ModelMeaslesMixingRiskQuarantine (
    ModelMeaslesMixingRiskQuarantine< TSeq > & model,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double vax_efficiency,
    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_high,
    epiworld_fast_int quarantine_period_medium,
    epiworld_fast_int quarantine_period_low,
    epiworld_double quarantine_willingness,
    epiworld_double isolation_willingness,
    epiworld_fast_int isolation_period,
    epiworld_double prop_vaccinated,
    epiworld_double detection_rate_quarantine,
    epiworld_double contact_tracing_success_rate = 1.0,
    epiworld_fast_uint contact_tracing_days_prior = 4u ) [inline]
```

Constructs a [ModelMeaslesMixingRiskQuarantine](#) object.

Parameters

<i>model</i>	A reference to an existing ModelMeaslesMixingRiskQuarantine 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_efficiency</i>	The efficacy of 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.

Parameters

<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_high</i>	The duration of quarantine in days for high-risk contacts.
<i>quarantine_period_medium</i>	The duration of quarantine in days for medium-risk contacts.
<i>quarantine_period_low</i>	The duration of quarantine in days for low-risk 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>detection_rate_quarantine</i>	The detection rate during active quarantine periods.
<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).

12.31.2.2 ModelMeaslesMixingRiskQuarantine() [2/2]

```
template<typename TSeq >
ModelMeaslesMixingRiskQuarantine< TSeq >::ModelMeaslesMixingRiskQuarantine (
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double vax_efficacy,
    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_high,
    epiworld_fast_int quarantine_period_medium,
    epiworld_fast_int quarantine_period_low,
    epiworld_double quarantine_willingness,
    epiworld_double isolation_willingness,
    epiworld_fast_int isolation_period,
    epiworld_double prop_vaccinated,
    epiworld_double detection_rate_quarantine,
    epiworld_double contact_tracing_success_rate = 1.0,
    epiworld_fast_uint contact_tracing_days_prior = 4u ) [inline]
```

Constructs a [ModelMeaslesMixingRiskQuarantine](#) 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.

Parameters

<i>vax_efficacy</i>	The efficacy of 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_high</i>	The duration of quarantine in days for high-risk contacts.
<i>quarantine_period_medium</i>	The duration of quarantine in days for medium-risk contacts.
<i>quarantine_period_low</i>	The duration of quarantine in days for low-risk 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>detection_rate_quarantine</i>	The detection rate during active quarantine periods.
<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).

12.31.3 Member Function Documentation**12.31.3.1 clone_ptr()**

```
template<typename TSeq >
Model< TSeq > * ModelMeaslesMixingRiskQuarantine< 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 >](#).

12.31.3.2 get_contact_matrix()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
auto epiworld::epimodels::ModelMeaslesMixingRiskQuarantine< TSeq >::get_contact_matrix ( )
const [inline]
```

Get the current contact matrix.

Returns

Vector representing the contact matrix

12.31.3.3 get_days_quarantine_triggered()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
auto epiworld::epimodels::ModelMeaslesMixingRiskQuarantine< TSeq >::get_days_quarantine_triggered() const [inline]
```

Get the total number of quarantines that have occurred.

Returns

The number of quarantines

12.31.3.4 get_isolation_willingness()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
auto epiworld::epimodels::ModelMeaslesMixingRiskQuarantine< 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

12.31.3.5 get_quarantine_risk_level()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
const auto& epiworld::epimodels::ModelMeaslesMixingRiskQuarantine< TSeq >::get_quarantine_risk_level() const [inline]
```

Get the risk level assigned to each agent for quarantine purposes.

Returns

Vector of integers representing risk levels (0=low, 1=medium, 2=high)

12.31.3.6 get_quarantine_willingness()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
auto epiworld::epimodels::ModelMeaslesMixingRiskQuarantine< 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

12.31.3.7 initial_states()

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

Set the initial states of the model.

Parameters

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

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

12.31.3.8 run()

```
template<typename TSeq >
ModelMeaslesMixingRiskQuarantine< TSeq > & ModelMeaslesMixingRiskQuarantine< 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 >](#).

12.31.3.9 set_contact_matrix()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
void epiworld::epimodels::ModelMeaslesMixingRiskQuarantine< 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:

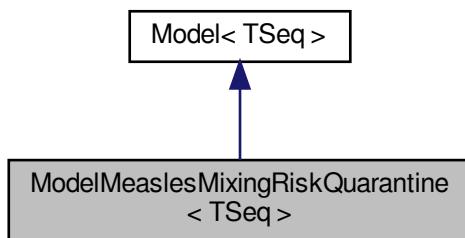
- epiworld.hpp

12.32 ModelMeaslesMixingRiskQuarantine< TSeq > Class Template Reference

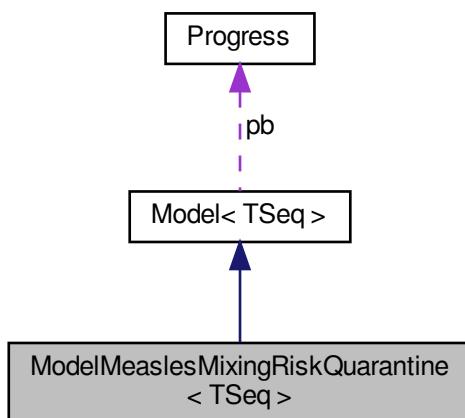
Measles model with population mixing and risk-based quarantine strategies.

```
#include <measlesmixingriskquarantine.hpp>
```

Inheritance diagram for ModelMeaslesMixingRiskQuarantine< TSeq >:



Collaboration diagram for ModelMeaslesMixingRiskQuarantine< TSeq >:



Public Member Functions

- [`ModelMeaslesMixingRiskQuarantine`](#) (`ModelMeaslesMixingRiskQuarantine< TSeq >` &`model`, `epiworld< _fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double vax_efficacy, 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_high, epiworld_fast_int quarantine_period_medium, epiworld_fast_int quarantine_period_low, epiworld_double quarantine_willingness, epiworld_double isolation_willingness, epiworld_fast_int isolation_period, epiworld_double prop_vaccinated, epiworld_double detection_rate_quarantine, epiworld_double contact_tracing_success_rate=1.0, epiworld_fast_uint contact_tracing_days_prior=4u)

Constructs a ModelMeaslesMixingRiskQuarantine object.`
- [`ModelMeaslesMixingRiskQuarantine`](#) (`epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double vax_efficacy, 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_high, epiworld_fast_int quarantine_period_medium, epiworld_fast_int quarantine_period_low, epiworld_double quarantine_willingness, epiworld_double isolation_willingness, epiworld_double prop_vaccinated, epiworld_double detection_rate_quarantine, epiworld_double contact_tracing_success_rate=1.0, epiworld_fast_uint contact_tracing_days_prior=4u)`

Constructs a `ModelMeaslesMixingRiskQuarantine` object.
- [`ModelMeaslesMixingRiskQuarantine< 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.
- [`ModelMeaslesMixingRiskQuarantine< 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.
- `auto get_contact_matrix () const`

Get the current contact matrix.
- `auto get_quarantine_willingness () const`

Get the quarantine willingness for all agents.
- `auto get_isolation_willingness () const`

Get the isolation willingness for all agents.
- `const auto & get_quarantine_risk_level () const`

Get the risk level assigned to each agent for quarantine purposes.
- `auto get_days_quarantine_triggered () const`

Get the total number of quarantines that have occurred.

Static Public Attributes

- `static constexpr int SUSCEPTIBLE = 0`
- `static constexpr int EXPOSED = 1`
- `static constexpr int PRODRMAL = 2`
- `static constexpr int RASH = 3`
- `static constexpr int ISOLATED = 4`
- `static constexpr int ISOLATED_RECOVERED = 5`
- `static constexpr int DETECTED_HOSPITALIZED = 6`

- static constexpr int **QUARANTINED_EXPOSED** = 7
- static constexpr int **QUARANTINED_SUSCEPTIBLE** = 8
- static constexpr int **QUARANTINED_PRODROMAL** = 9
- static constexpr int **QUARANTINED_RECOVERED** = 10
- static constexpr int **HOSPITALIZED** = 11
- static constexpr int **RECOVERED** = 12
- static constexpr size_t **QUARANTINE_PROCESS_INACTIVE** = 0u
- static constexpr size_t **QUARANTINE_PROCESS_ACTIVE** = 1u
- static constexpr size_t **QUARANTINE_PROCESS_DONE** = 2u
- static constexpr int **RISK_LOW** = 0
- static constexpr int **RISK_MEDIUM** = 1
- static constexpr int **RISK_HIGH** = 2

Additional Inherited Members

12.32.1 Detailed Description

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

Measles model with population mixing and risk-based quarantine strategies.

This class extends the Measles epidemiological model to support different quarantine strategies based on exposure risk levels:

- **High Risk:** Unvaccinated agents who share entity membership with the case
- **Medium Risk:** Unvaccinated agents who contacted an infected individual but don't share entity membership
- **Low Risk:** Other unvaccinated agents

Each risk level can have different quarantine durations, allowing for targeted public health interventions. The model also includes enhanced detection during active quarantine periods.

Disease progression follows the same states as [ModelMeaslesMixing](#): Susceptible → Exposed → Prodromal → Rash → Recovered

Template Parameters

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

12.32.2 Constructor & Destructor Documentation

12.32.2.1 ModelMeaslesMixingRiskQuarantine() [1/2]

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

```
ModelMeaslesMixingRiskQuarantine< TSeq > & model,
epiworld_fast_uint n,
epiworld_double prevalence,
epiworld_double contact_rate,
epiworld_double transmission_rate,
epiworld_double vax_efficacy,
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_high,
epiworld_fast_int quarantine_period_medium,
epiworld_fast_int quarantine_period_low,
epiworld_double quarantine_willingness,
epiworld_double isolation_willingness,
epiworld_fast_int isolation_period,
epiworld_double prop_vaccinated,
epiworld_double detection_rate_quarantine,
epiworld_double contact_tracing_success_rate = 1.0,
epiworld_fast_uint contact_tracing_days_prior = 4u ) [inline]
```

Constructs a [ModelMeaslesMixingRiskQuarantine](#) object.

Parameters

<i>model</i>	A reference to an existing ModelMeaslesMixingRiskQuarantine 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>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_high</i>	The duration of quarantine in days for high-risk contacts.
<i>quarantine_period_medium</i>	The duration of quarantine in days for medium-risk contacts.
<i>quarantine_period_low</i>	The duration of quarantine in days for low-risk 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>detection_rate_quarantine</i>	The detection rate during active quarantine periods.
<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).

12.32.2.2 ModelMeaslesMixingRiskQuarantine() [2/2]

```
template<typename TSeq >
ModelMeaslesMixingRiskQuarantine< TSeq >::ModelMeaslesMixingRiskQuarantine (
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double vax_efficacy,
    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_high,
    epiworld_fast_int quarantine_period_medium,
    epiworld_fast_int quarantine_period_low,
    epiworld_double quarantine_willingness,
    epiworld_double isolation_willingness,
    epiworld_fast_int isolation_period,
    epiworld_double prop_vaccinated,
    epiworld_double detection_rate_quarantine,
    epiworld_double contact_tracing_success_rate = 1.0,
    epiworld_fast_uint contact_tracing_days_prior = 4u ) [inline]
```

Constructs a [ModelMeaslesMixingRiskQuarantine](#) 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>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_high</i>	The duration of quarantine in days for high-risk contacts.
<i>quarantine_period_medium</i>	The duration of quarantine in days for medium-risk contacts.
<i>quarantine_period_low</i>	The duration of quarantine in days for low-risk 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>detection_rate_quarantine</i>	The detection rate during active quarantine periods.
<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).

12.32.3 Member Function Documentation

12.32.3.1 clone_ptr()

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

Create a clone of this model.

Returns

Pointer to a new model instance with the same configuration

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

12.32.3.2 get_contact_matrix()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
auto ModelMeaslesMixingRiskQuarantine< TSeq >::get_contact_matrix ( ) const [inline]
```

Get the current contact matrix.

Returns

Vector representing the contact matrix

12.32.3.3 get_days_quarantine_triggered()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
auto ModelMeaslesMixingRiskQuarantine< TSeq >::get_days_quarantine_triggered ( ) const [inline]
```

Get the total number of quarantines that have occurred.

Returns

The number of quarantines

12.32.3.4 get_isolation_willingness()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
auto ModelMeaslesMixingRiskQuarantine< 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

12.32.3.5 get_quarantine_risk_level()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
const auto& ModelMeaslesMixingRiskQuarantine< TSeq >::get_quarantine_risk_level ( ) const [inline]
```

Get the risk level assigned to each agent for quarantine purposes.

Returns

Vector of integers representing risk levels (0=low, 1=medium, 2=high)

12.32.3.6 get_quarantine_willingness()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
auto ModelMeaslesMixingRiskQuarantine< 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

12.32.3.7 initial_states()

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

Set the initial states of the model.

Parameters

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

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

12.32.3.8 run()

```
template<typename TSeq >
ModelMeaslesMixingRiskQuarantine< TSeq > & ModelMeaslesMixingRiskQuarantine< 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 [Model< TSeq >](#).

12.32.3.9 set_contact_matrix()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
void ModelMeaslesMixingRiskQuarantine< 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:

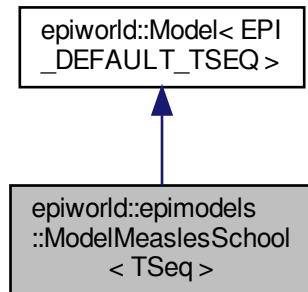
- include/epiworld/models/[measlesmixingriskquarantine.hpp](#)

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

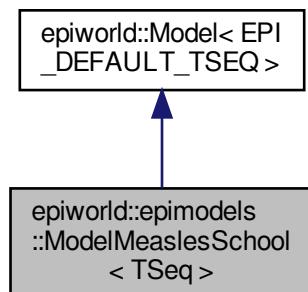
Template for a Measles model with quarantine.

```
#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 PRODRONAL = 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

12.33.1 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.

12.33.2 Constructor & Destructor Documentation

12.33.2.1 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_efficiency,
    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_efficiency</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.

12.33.3 Member Function Documentation**12.33.3.1 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>	<input type="checkbox"/>
-------------	--------------------------

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

12.33.3.2 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.

12.33.3.3 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:

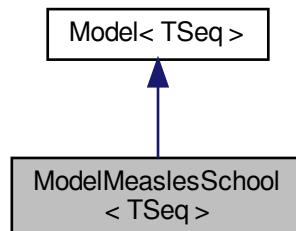
- `epiworld.hpp`

12.34 ModelMeaslesSchool< TSeq > Class Template Reference

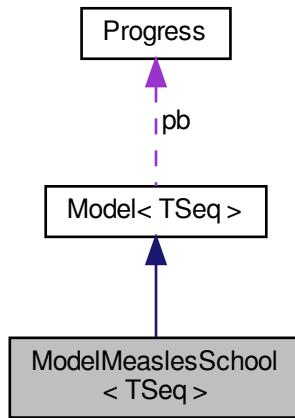
Template for a Measles model with quarantine.

```
#include <measlesschool.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

12.34.1 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.

12.34.2 Constructor & Destructor Documentation

12.34.2.1 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.

12.34.3 Member Function Documentation**12.34.3.1 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>	<input type="checkbox"/>
-------------	--------------------------

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

12.34.3.2 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_*
- At the end of the function, the quarantine status is set false.

12.34.3.3 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:

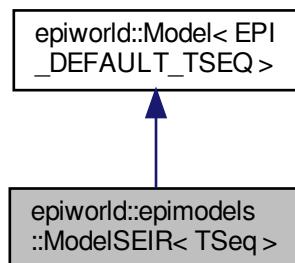
- include/epiworld/models/measlesschool.hpp

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

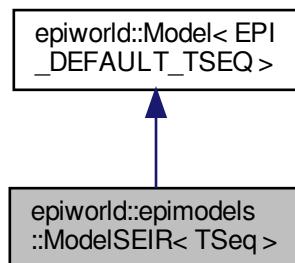
Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.

```
#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)`)
- **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

12.35.1 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.

12.35.2 Member Function Documentation

12.35.2.1 initial_states()

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

Set up the initial states of the model.

Parameters

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

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

12.35.3 Member Data Documentation

12.35.3.1 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;
}
```

12.35.3.2 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:

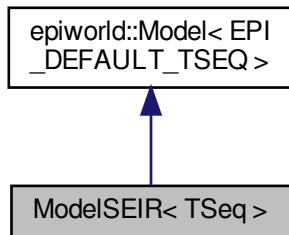
- [epiworld.hpp](#)

12.36 ModelSEIR< TSeq > Class Template Reference

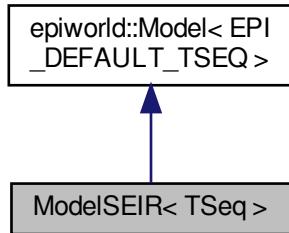
Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.

```
#include <seir.hpp>
```

Inheritance diagram for ModelSEIR< TSeq >:



Collaboration diagram for ModelSEIR< TSeq >:



Public Member Functions

- **ModelSEIR** (`ModelSEIR< TSeq > &model, const std::string &vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate)`)
- **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

12.36.1 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.

12.36.2 Member Function Documentation

12.36.2.1 initial_states()

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

Set up the initial states of the model.

Parameters

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

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

12.36.3 Member Data Documentation

12.36.3.1 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;
}
```

12.36.3.2 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 {

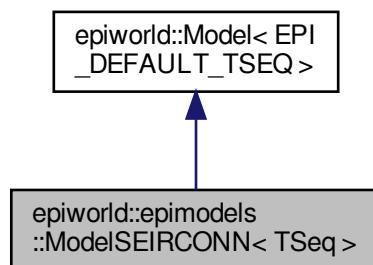
    if (m->runif() < (m->par("Recovery rate")))
        p->rm_virus(m);
    return;
}
```

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

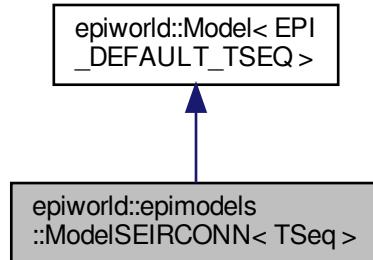
- include/epiworld/models/seir.hpp

12.37 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

12.37.1 Constructor & Destructor Documentation

12.37.1.1 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

12.37.2 Member Function Documentation

12.37.2.1 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>	<input type="checkbox"/>
-------------	--------------------------

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

12.37.2.2 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: – The proportion of non-infected individuals who have recovered.
----------------------	--

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

12.37.2.3 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

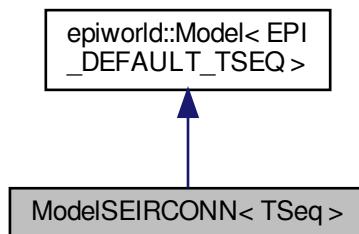
Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

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

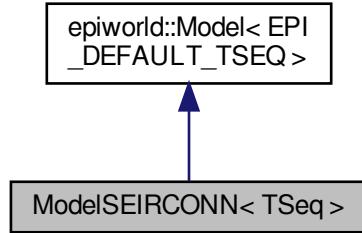
- `epiworld.hpp`

12.38 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

12.38.1 Constructor & Destructor Documentation

12.38.1.1 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

12.38.2 Member Function Documentation

12.38.2.1 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>	<input type="checkbox"/>
-------------	--------------------------

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

12.38.2.2 initial_states()

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

Set the initial states of the model.

Parameters

<i>proportions</i> ←	Double vector with a single element:
—	<ul style="list-style-type: none"> The proportion of non-infected individuals who have recovered.

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

12.38.2.3 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:

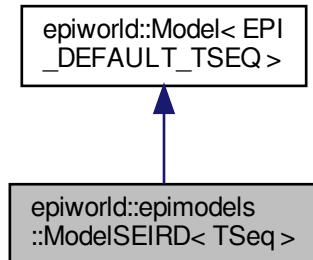
- include/epiworld/models/seirconnected.hpp

12.39 epiworld::epimodels::ModelSEIRD< TSeq > Class Template Reference

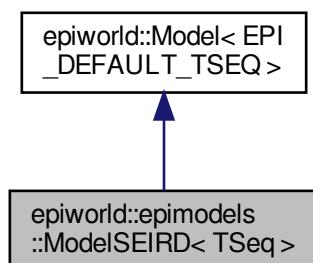
Template for a Susceptible-Exposed-Infected-Removed-Deceased (SEIRD) model.

```
#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

12.39.1 Detailed Description

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

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

12.39.2 Constructor & Destructor Documentation

12.39.2.1 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.
<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.

12.39.2.2 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.

12.39.3 Member Data Documentation

12.39.3.1 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:

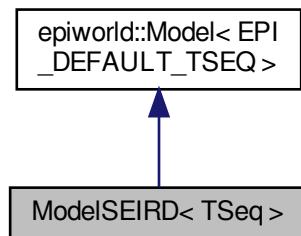
- epiworld.hpp

12.40 ModelSEIRD< TSeq > Class Template Reference

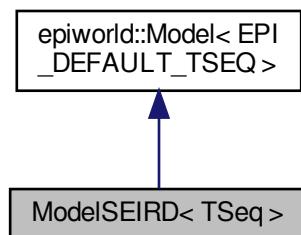
Template for a Susceptible-Exposed-Infected-Removed-Deceased (SEIRD) model.

```
#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

12.40.1 Detailed Description

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

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

12.40.2 Constructor & Destructor Documentation

12.40.2.1 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.

12.40.2.2 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.

12.40.3 Member Data Documentation**12.40.3.1 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();

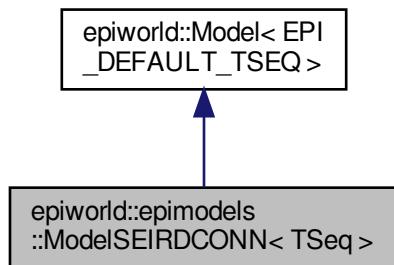
    if (m->runif() < 1.0/(v->get_incubation(m)))
        p->change_state(m, ModelSEIRD<TSeq>::INFECTED);
    return;
}
```

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

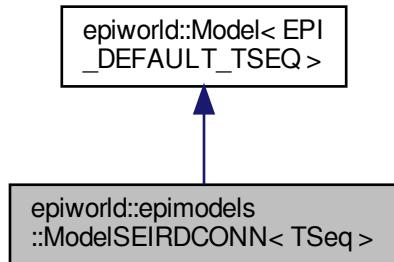
- include/epiworld/models/seird.hpp

12.41 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

12.41.1 Constructor & Destructor Documentation

12.41.1.1 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

12.41.2 Member Function Documentation

12.41.2.1 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 >](#).

12.41.2.2 initial_states()

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

Set up the initial states of the model.

Parameters

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

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

12.41.2.3 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

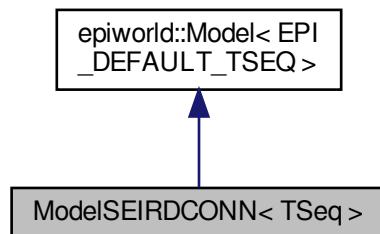
Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

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

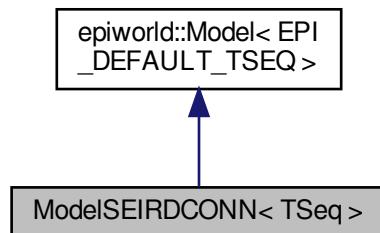
- `epiworld.hpp`

12.42 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

12.42.1 Constructor & Destructor Documentation

12.42.1.1 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

12.42.2 Member Function Documentation**12.42.2.1 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 >](#).

12.42.2.2 initial_states()

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

Set up the initial states of the model.

Parameters

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

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

12.42.2.3 `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

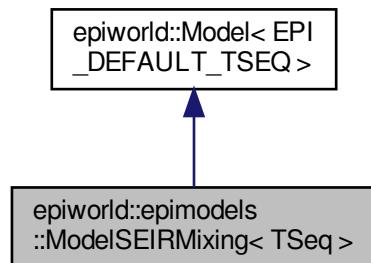
Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

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

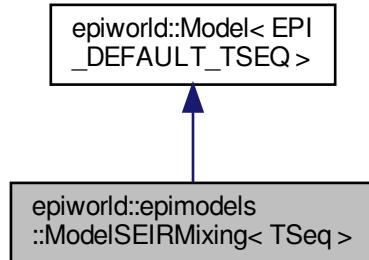
- `include/epiworld/models/seirdconnected.hpp`

12.43 `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

12.43.1 Constructor & Destructor Documentation

12.43.1.1 ModelSEIRMixing() [1/2]

```
template<typename TSeq >
ModelSEIRMixing< TSeq >::ModelSEIRMixing (
    ModelSEIRMixing< TSeq > & model,
    const std::string & vname,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double avg_incubation_days,
    epiworld_double recovery_rate,
    std::vector< double > contact_matrix ) [inline]
```

Constructs a [ModelSEIRMixing](#) object.

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

Parameters

<i>model</i>	A reference to an existing ModelSEIRMixing object.
<i>vname</i>	The name of the ModelSEIRMixing object.
<i>n</i>	The number of entities in the model.
<i>prevalence</i>	The initial prevalence of the disease in the model.
<i>contact_rate</i>	The contact rate between entities in the model.
<i>transmission_rate</i>	The transmission rate of the disease in the model.
<i>avg_incubation_days</i>	The average incubation period of the disease in the model.
<i>recovery_rate</i>	The recovery rate of the disease in the model.
<i>contact_matrix</i>	The contact matrix between entities in the model. Specified in column-major order.
<i>model</i>	A Model<TSeq> object where to set up the SIR.
<i>vname</i>	std::string Name of the virus
<i>prevalence</i>	Initial prevalence (proportion)
<i>contact_rate</i>	Average number of contacts (interactions) per step.
<i>transmission_rate</i>	Probability of transmission
<i>recovery_rate</i>	Probability of recovery

12.43.1.2 ModelSEIRMixing() [2/2]

```
template<typename TSeq >
ModelSEIRMixing< TSeq >::ModelSEIRMixing (
    const std::string & vname,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double avg_incubation_days,
    epiworld_double recovery_rate,
    std::vector< double > contact_matrix ) [inline]
```

Constructs a [ModelSEIRMixing](#) object.

Parameters

<i>vname</i>	The name of the ModelSEIRMixing object.
<i>n</i>	The number of entities in the model.
<i>prevalence</i>	The initial prevalence of the disease in the model.
<i>contact_rate</i>	The contact rate between entities in the model.
<i>transmission_rate</i>	The transmission rate of the disease in the model.
<i>avg_incubation_days</i>	The average incubation period of the disease in the model.
<i>recovery_rate</i>	The recovery rate of the disease in the model.
<i>contact_matrix</i>	The contact matrix between entities in the model.

12.43.2 Member Function Documentation**12.43.2.1 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 >](#).

12.43.2.2 initial_states()

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

Set the initial states of the model.

Parameters

<i>proportions</i> ↪	Double vector with a single element:
—	<ul style="list-style-type: none"> The proportion of non-infected individuals who have recovered.

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

12.43.2.3 `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

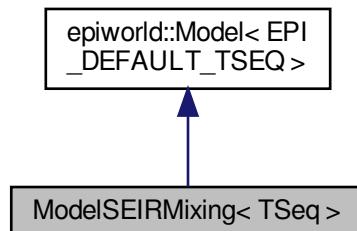
Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

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

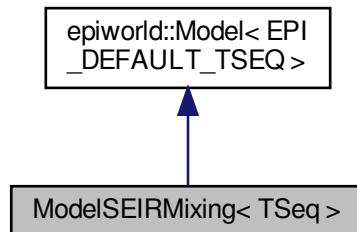
- `epiworld.hpp`

12.44 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

12.44.1 Constructor & Destructor Documentation

12.44.1.1 ModelSEIRMixing() [1/2]

```
template<typename TSeq >
ModelSEIRMixing< TSeq >::ModelSEIRMixing (
    ModelSEIRMixing< TSeq > & model,
    const std::string & vname,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double avg_incubation_days,
    epiworld_double recovery_rate,
    std::vector< double > contact_matrix ) [inline]
```

Constructs a `ModelSEIRMixing` object.

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

Parameters

<i>model</i>	A reference to an existing ModelSEIRMixing object.
<i>vname</i>	The name of the ModelSEIRMixing object.
<i>n</i>	The number of entities in the model.
<i>prevalence</i>	The initial prevalence of the disease in the model.
<i>contact_rate</i>	The contact rate between entities in the model.
<i>transmission_rate</i>	The transmission rate of the disease in the model.
<i>avg_incubation_days</i>	The average incubation period of the disease in the model.
<i>recovery_rate</i>	The recovery rate of the disease in the model.
<i>contact_matrix</i>	The contact matrix between entities in the model. Specified in column-major order.
<i>model</i>	A Model<TSeq> object where to set up the SIR.
<i>vname</i>	std::string Name of the virus
<i>prevalence</i>	Initial prevalence (proportion)
<i>contact_rate</i>	Average number of contacts (interactions) per step.
<i>transmission_rate</i>	Probability of transmission
<i>recovery_rate</i>	Probability of recovery

12.44.1.2 ModelSEIRMixing() [2/2]

```
template<typename TSeq >
ModelSEIRMixing< TSeq >::ModelSEIRMixing (
    const std::string & vname,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double avg_incubation_days,
    epiworld_double recovery_rate,
    std::vector< double > contact_matrix ) [inline]
```

Constructs a [ModelSEIRMixing](#) object.

Parameters

<i>vname</i>	The name of the ModelSEIRMixing object.
<i>n</i>	The number of entities in the model.
<i>prevalence</i>	The initial prevalence of the disease in the model.
<i>contact_rate</i>	The contact rate between entities in the model.
<i>transmission_rate</i>	The transmission rate of the disease in the model.
<i>avg_incubation_days</i>	The average incubation period of the disease in the model.
<i>recovery_rate</i>	The recovery rate of the disease in the model.
<i>contact_matrix</i>	The contact matrix between entities in the model.

12.44.2 Member Function Documentation

12.44.2.1 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 >](#).

12.44.2.2 initial_states()

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

Set the initial states of the model.

Parameters

<i>proportions</i> ↪	Double vector with a single element:
—	<ul style="list-style-type: none"> The proportion of non-infected individuals who have recovered.

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

12.44.2.3 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

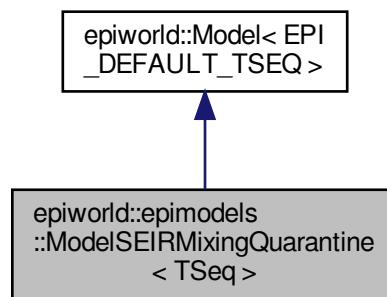
12.45 epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >

Class Template Reference

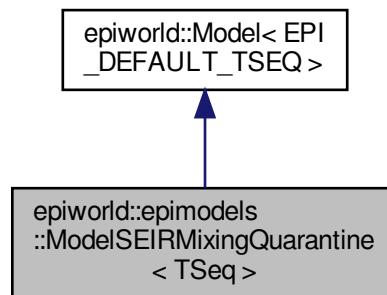
SEIR model with mixing, quarantine, and contact tracing.

```
#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

12.45.1 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

TSeq	Type for genetic sequences (default: EPI_DEFAULT_TSEQ)
------	--

12.45.2 Constructor & Destructor Documentation

12.45.2.1 ModelSEIRMixingQuarantine() [1/2]

```
template<typename TSeq >
ModelSEIRMixingQuarantine< TSeq >::ModelSEIRMixingQuarantine (
    ModelSEIRMixingQuarantine< TSeq > & model,
    const std::string & vname,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double avg_incubation_days,
    epiworld_double recovery_rate,
    std::vector< double > contact_matrix,
    epiworld_double hospitalization_rate,
    epiworld_double hospitalization_period,
    epiworld_double days_undetected,
    epiworld_fast_int quarantine_period,
    epiworld_double quarantine_willingness,
    epiworld_double isolation_willingness,
    epiworld_fast_int isolation_period,
    epiworld_double contact_tracing_success_rate = 1.0,
    epiworld_fast_uint contact_tracing_days_prior = 4u ) [inline]
```

Constructs a [ModelSEIRMixingQuarantine](#) object.

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

Parameters

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

Parameters

<i>contact_rate</i>	Average number of contacts (interactions) per step
<i>transmission_rate</i>	Probability of transmission per contact
<i>avg_incubation_days</i>	Average incubation period in days
<i>recovery_rate</i>	Probability of recovery per day
<i>contact_matrix</i>	Contact matrix specifying mixing patterns between population groups
<i>hospitalization_rate</i>	Rate at which infected individuals are hospitalized
<i>hospitalization_period</i>	Average duration of hospitalization in days
<i>days_undetected</i>	Average number of days an infected individual remains undetected
<i>quarantine_period</i>	Duration of quarantine in days for exposed contacts
<i>quarantine_willingness</i>	Proportion of individuals willing to comply with quarantine
<i>isolation_willingness</i>	Proportion of individuals willing to self-isolate when detected
<i>isolation_period</i>	Duration of isolation in days for detected infected individuals
<i>contact_tracing_success_rate</i>	Probability of successfully identifying contacts during tracing
<i>contact_tracing_days_prior</i>	Number of days prior to detection for contact tracing

12.45.2.2 ModelSEIRMixingQuarantine() [2/2]

```
template<typename TSeq >
ModelSEIRMixingQuarantine< TSeq >::ModelSEIRMixingQuarantine (
    const std::string & vname,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double avg_incubation_days,
    epiworld_double recovery_rate,
    std::vector< double > contact_matrix,
    epiworld_double hospitalization_rate,
    epiworld_double hospitalization_period,
    epiworld_double days_undetected,
    epiworld_fast_int quarantine_period,
    epiworld_double quarantine_willingness,
    epiworld_double isolation_willingness,
    epiworld_fast_int isolation_period,
    epiworld_double contact_tracing_success_rate = 1.0,
    epiworld_fast_uint contact_tracing_days_prior = 4u ) [inline]
```

Constructs a [ModelSEIRMixingQuarantine](#) object.

Parameters

<i>vname</i>	The name of the ModelSEIRMixingQuarantine object.
<i>n</i>	The number of entities in the model.
<i>prevalence</i>	The initial prevalence of the disease in the model.
<i>contact_rate</i>	The contact rate between entities in the model.
<i>transmission_rate</i>	The transmission rate of the disease in the model.
<i>avg_incubation_days</i>	The average incubation period of the disease in the model.

Parameters

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

12.45.3 Member Function Documentation**12.45.3.1 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 >](#).

12.45.3.2 get_agent_quarantine_triggered()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
std::vector< size_t > epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >::get_agent_triggered ( ) const [inline]
```

Get the quarantine trigger status for all agents.

Returns

Vector indicating quarantine process status for each agent

12.45.3.3 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

12.45.3.4 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

12.45.3.5 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

12.45.3.6 initial_states()

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

Set the initial states of the model.

Parameters

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

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

12.45.3.7 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 >](#).

12.45.3.8 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:

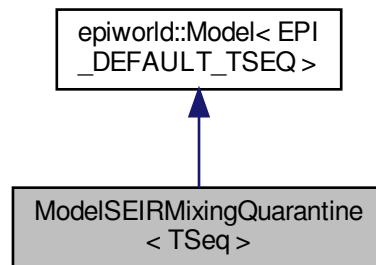
- epiworld.hpp

12.46 ModelSEIRMixingQuarantine< TSeq > Class Template Reference

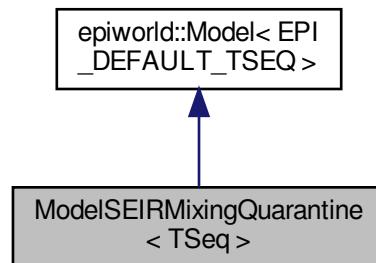
SEIR model with mixing, quarantine, and contact tracing.

```
#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

12.46.1 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

TSeq	Type for genetic sequences (default: EPI_DEFAULT_TSEQ)
------	--

12.46.2 Constructor & Destructor Documentation

12.46.2.1 ModelSEIRMixingQuarantine() [1/2]

```
template<typename TSeq >
ModelSEIRMixingQuarantine< TSeq >::ModelSEIRMixingQuarantine (
    ModelSEIRMixingQuarantine< TSeq > & model,
    const std::string & vname,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double avg_incubation_days,
    epiworld_double recovery_rate,
    std::vector< double > contact_matrix,
    epiworld_double hospitalization_rate,
    epiworld_double hospitalization_period,
    epiworld_double days_undetected,
    epiworld_fast_int quarantine_period,
    epiworld_double quarantine_willingness,
    epiworld_double isolation_willingness,
    epiworld_fast_int isolation_period,
    epiworld_double contact_tracing_success_rate = 1.0,
    epiworld_fast_uint contact_tracing_days_prior = 4u ) [inline]
```

Constructs a [ModelSEIRMixingQuarantine](#) object.

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

Parameters

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

Parameters

<i>contact_rate</i>	Average number of contacts (interactions) per step
<i>transmission_rate</i>	Probability of transmission per contact
<i>avg_incubation_days</i>	Average incubation period in days
<i>recovery_rate</i>	Probability of recovery per day
<i>contact_matrix</i>	Contact matrix specifying mixing patterns between population groups
<i>hospitalization_rate</i>	Rate at which infected individuals are hospitalized
<i>hospitalization_period</i>	Average duration of hospitalization in days
<i>days_undetected</i>	Average number of days an infected individual remains undetected
<i>quarantine_period</i>	Duration of quarantine in days for exposed contacts
<i>quarantine_willingness</i>	Proportion of individuals willing to comply with quarantine
<i>isolation_willingness</i>	Proportion of individuals willing to self-isolate when detected
<i>isolation_period</i>	Duration of isolation in days for detected infected individuals
<i>contact_tracing_success_rate</i>	Probability of successfully identifying contacts during tracing
<i>contact_tracing_days_prior</i>	Number of days prior to detection for contact tracing

12.46.2.2 ModelSEIRMixingQuarantine() [2/2]

```
template<typename TSeq >
ModelSEIRMixingQuarantine< TSeq >::ModelSEIRMixingQuarantine (
    const std::string & vname,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double avg_incubation_days,
    epiworld_double recovery_rate,
    std::vector< double > contact_matrix,
    epiworld_double hospitalization_rate,
    epiworld_double hospitalization_period,
    epiworld_double days_undetected,
    epiworld_fast_int quarantine_period,
    epiworld_double quarantine_willingness,
    epiworld_double isolation_willingness,
    epiworld_fast_int isolation_period,
    epiworld_double contact_tracing_success_rate = 1.0,
    epiworld_fast_uint contact_tracing_days_prior = 4u ) [inline]
```

Constructs a [ModelSEIRMixingQuarantine](#) object.

Parameters

<i>vname</i>	The name of the ModelSEIRMixingQuarantine object.
<i>n</i>	The number of entities in the model.
<i>prevalence</i>	The initial prevalence of the disease in the model.
<i>contact_rate</i>	The contact rate between entities in the model.
<i>transmission_rate</i>	The transmission rate of the disease in the model.
<i>avg_incubation_days</i>	The average incubation period of the disease in the model.

Parameters

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

12.46.3 Member Function Documentation**12.46.3.1 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 >](#).

12.46.3.2 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

12.46.3.3 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

12.46.3.4 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

12.46.3.5 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

12.46.3.6 initial_states()

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

Set the initial states of the model.

Parameters

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

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

12.46.3.7 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 >](#).

12.46.3.8 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:

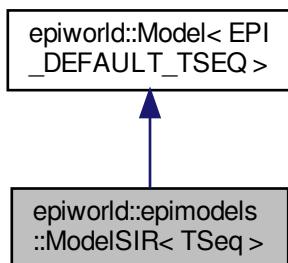
- include/epiworld/models/[seirmixingquarantine.hpp](#)

12.47 epiworld::epimodels::ModelSIR< TSeq > Class Template Reference

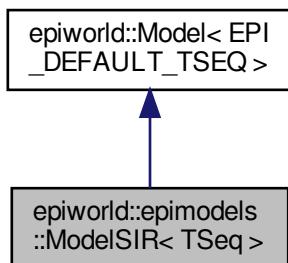
Template for a Susceptible-Infected-Removed (SIR) model.

```
#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

12.47.1 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_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

12.47.2 Member Function Documentation

12.47.2.1 initial_states()

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

Set the initial states of the model.

Parameters

<i>proportions</i> ↪	Double vector with a single element:
—	<ul style="list-style-type: none"> The proportion of non-infected individuals who have recovered.

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

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

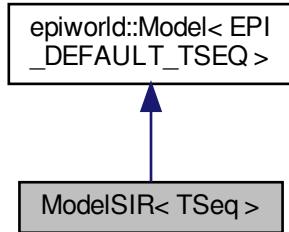
- [epiworld.hpp](#)

12.48 ModelSIR< TSeq > Class Template Reference

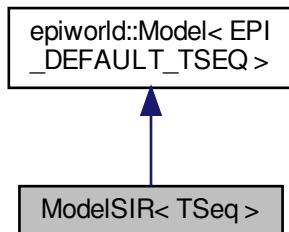
Template for a Susceptible-Infected-Removed (SIR) model.

```
#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

12.48.1 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

12.48.2 Member Function Documentation

12.48.2.1 initial_states()

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

Set the initial states of the model.

Parameters

<i>proportions</i> ←	Double vector with a single element:
—	<ul style="list-style-type: none"> The proportion of non-infected individuals who have recovered.

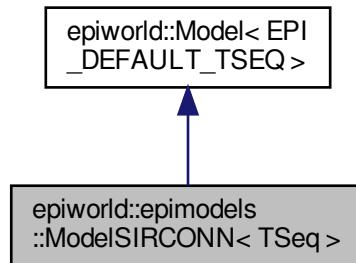
Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

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

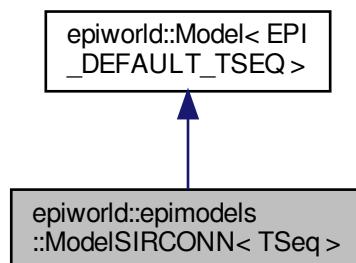
- include/epiworld/models/sir.hpp

12.49 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

12.49.1 Constructor & Destructor Documentation

12.49.1.1 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 <code>Model<TSeq></code> object where to set up the SIR.
<i>vname</i>	<code>std::string</code> Name of the virus
<i>prevalence</i>	Initial prevalence (proportion)
<i>contact_rate</i>	Average number of contacts (interactions) per step.
<i>transmission_rate</i>	Probability of transmission
<i>recovery_rate</i>	Probability of recovery

12.49.2 Member Function Documentation

12.49.2.1 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 >](#).

12.49.2.2 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> * >
```

12.49.2.3 initial_states()

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

Set the initial states of the model.

Parameters

<i>proportions</i> ↗	Double vector with a single element:
—	<ul style="list-style-type: none"> The proportion of non-infected individuals who have recovered.

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

12.49.2.4 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

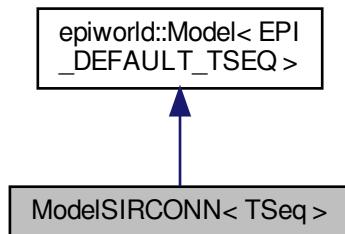
Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

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

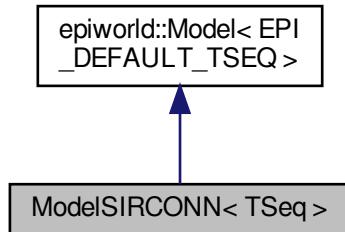
- `epiworld.hpp`

12.50 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

12.50.1 Constructor & Destructor Documentation

12.50.1.1 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 <code>Model<TSeq></code> 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

12.50.2 Member Function Documentation

12.50.2.1 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 >](#).

12.50.2.2 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> * >`

12.50.2.3 initial_states()

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

Set the initial states of the model.

Parameters

<i>proportions</i> ←	Double vector with a single element:
—	<ul style="list-style-type: none"> The proportion of non-infected individuals who have recovered.

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

12.50.2.4 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:

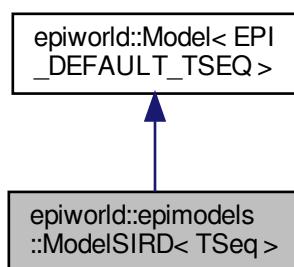
- `include/epiworld/models/sirconnected.hpp`

12.51 epiworld::epimodels::ModelSIRD< TSeq > Class Template Reference

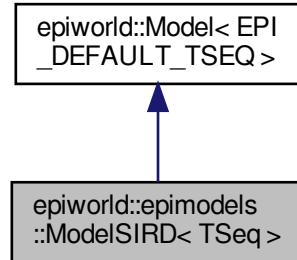
Template for a Susceptible-Infected-Removed-Deceased (SIRD) model.

```
#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

12.51.1 Detailed Description

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

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

12.51.2 Constructor & Destructor Documentation

12.51.2.1 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.

12.51.3 Member Function Documentation**12.51.3.1 initial_states()**

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

Set the initial states of the model.

Parameters

<i>proportions</i> ↪	Double vector with two elements:
–	<ul style="list-style-type: none"> The proportion of non-infected individuals who have recovered. The proportion of non-infected individuals who have died.

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

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

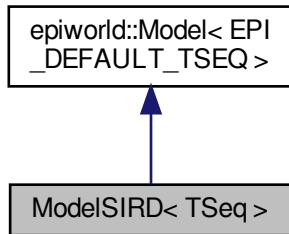
- epiworld.hpp

12.52 ModelSIRD< TSeq > Class Template Reference

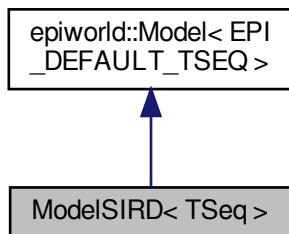
Template for a Susceptible-Infected-Removed-Deceased (SIRD) model.

```
#include <sird.hpp>
```

Inheritance diagram for ModelSIRD< TSeq >:



Collaboration diagram for ModelSIRD< TSeq >:



Public Member Functions

- [ModelSIRD< TSeq > & initial_states \(std::vector< double > proportions_, std::vector< int > queue_={}\)](#)
Set the initial states of the model.
- [ModelSIRD \(ModelSIRD< TSeq > &model, const std::string &vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double recovery_rate, epiworld_double death_rate\)](#)
Constructs a new SIRD model with the given parameters.
- [ModelSIRD \(const std::string &vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double recovery_rate, epiworld_double death_rate\)](#)

Additional Inherited Members

12.52.1 Detailed Description

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

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

12.52.2 Constructor & Destructor Documentation

12.52.2.1 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.

12.52.3 Member Function Documentation

12.52.3.1 initial_states()

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

Set the initial states of the model.

Parameters

<i>proportions</i> ↗	Double vector with two elements: <ul style="list-style-type: none"> – The proportion of non-infected individuals who have recovered. – The proportion of non-infected individuals who have died.
----------------------	--

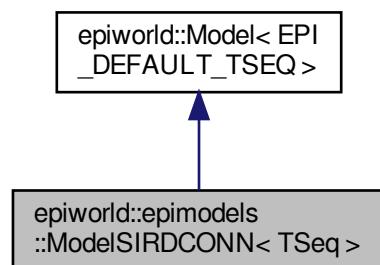
Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

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

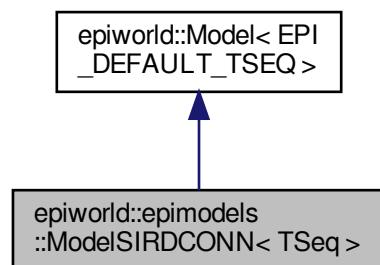
- include/epiworld/models/sird.hpp

12.53 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

12.53.1 Constructor & Destructor Documentation

12.53.1.1 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

12.53.2 Member Function Documentation

12.53.2.1 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 >](#).

12.53.2.2 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

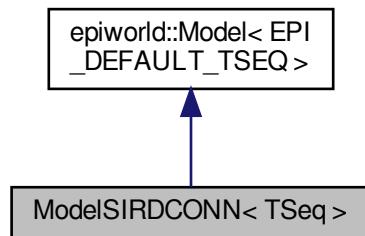
Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

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

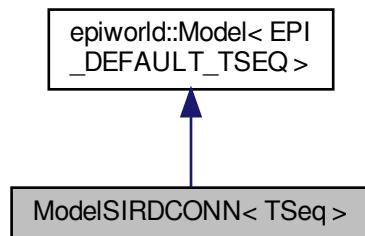
- `epiworld.hpp`

12.54 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

12.54.1 Constructor & Destructor Documentation

12.54.1.1 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

12.54.2 Member Function Documentation

12.54.2.1 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>	<input type="button" value=""/>
-------------	---------------------------------

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

12.54.2.2 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:

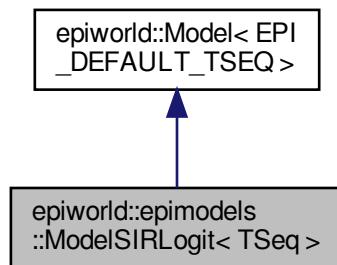
- include/epiworld/models/sirdconnected.hpp

12.55 epiworld::epimodels::ModelSIRLogit< TSeq > Class Template Reference

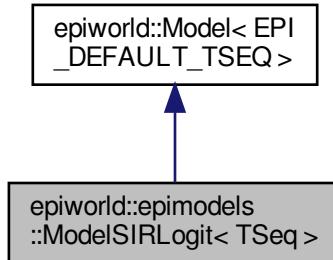
Template for a Susceptible-Infected-Removed (SIR) model.

```
#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

12.55.1 Detailed Description

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

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

In this model, infection and recovery probabilities are computed using a logit model. Particularly, the probability of infection is computed as:

$$\frac{1}{1 + \exp(-(\beta_0 E_i + \sum_{i=1}^n \beta_i x_i))}$$

where β_0 is the exposure coefficient and E_i is the exposure number, β_i are the coefficients for the features x_i of the agents, and n is the number of features. The probability of recovery is computed as:

$$\frac{1}{1 + \exp(-(\sum_{i=1}^n \beta_i x_i))}$$

where β_i are the coefficients for the features x_i of the agents, and n is the number of features.

Parameters

TSeq	Type of the sequence (e.g. std::vector, std::deque)
------	---

12.55.2 Constructor & Destructor Documentation

12.55.2.1 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

vname	Name of the virus.
coefs_infect	Double ptr. Infection coefficients.
coefs_recover	Double ptr. Recovery coefficients.
ncoef_infect	Unsigned int. Number of infection coefficients.
ncoef_recover	Unsigned int. Number of recovery coefficients.
coef_infect_cols	Vector<unsigned int>. Ids of infection vars.
coef_recover_cols	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

12.55.3 Member Function Documentation**12.55.3.1 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>	<input type="checkbox"/>
-------------	--------------------------

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

12.55.3.2 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:

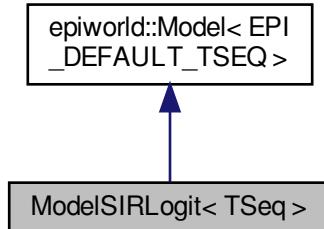
- `epiworld.hpp`

12.56 ModelSIRLogit< TSeq > Class Template Reference

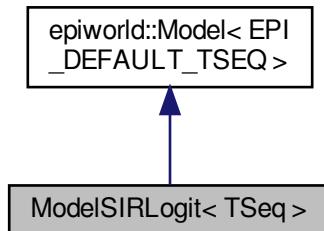
Template for a Susceptible-Infected-Removed (SIR) model.

```
#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

12.56.1 Detailed Description

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

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

In this model, infection and recovery probabilities are computed using a logit model. Particularly, the probability of infection is computed as:

$$\frac{1}{1 + \exp(-(\beta_0 E_i + \sum_{i=1}^n \beta_i x_i))}$$

where β_0 is the exposure coefficient and E_i is the exposure number, β_i are the coefficients for the features x_i of the agents, and n is the number of features. The probability of recovery is computed as:

$$\frac{1}{1 + \exp(-(\sum_{i=1}^n \beta_i x_i))}$$

where β_i are the coefficients for the features x_i of the agents, and n is the number of features.

Parameters

<i>TSeq</i>	Type of the sequence (e.g. std::vector, std::deque)
-------------	---

12.56.2 Constructor & Destructor Documentation

12.56.2.1 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

12.56.3 Member Function Documentation

12.56.3.1 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 >](#).

12.56.3.2 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

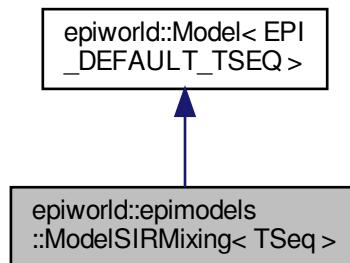
Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

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

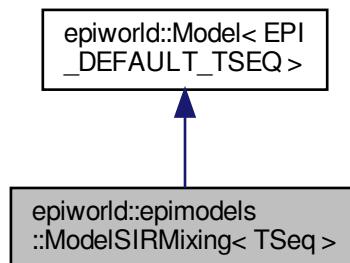
- include/epiworld/models/sirlogit.hpp

12.57 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

12.57.1 Constructor & Destructor Documentation

12.57.1.1 ModelSIRMixing() [1/2]

```
template<typename TSeq >
ModelSIRMixing< TSeq >::ModelSIRMixing (
    ModelSIRMixing< TSeq > & model,
    const std::string & vname,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double recovery_rate,
    std::vector< double > contact_matrix ) [inline]
```

Constructs a ModelSIRMixing object.

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

Parameters

<i>model</i>	A reference to an existing ModelSIRMixing object.
<i>vname</i>	The name of the ModelSIRMixing object.
<i>n</i>	The number of entities in the model.
<i>prevalence</i>	The initial prevalence of the disease in the model.
<i>contact_rate</i>	The contact rate between entities in the model.
<i>transmission_rate</i>	The transmission rate of the disease in the model.
<i>recovery_rate</i>	The recovery rate of the disease in the model.
<i>contact_matrix</i>	The contact matrix between entities in the model.
<i>model</i>	A Model<TSeq> object where to set up the SIR.
<i>vname</i>	std::string Name of the virus
<i>prevalence</i>	Initial prevalence (proportion)
<i>contact_rate</i>	Average number of contacts (interactions) per step.
<i>transmission_rate</i>	Probability of transmission
<i>recovery_rate</i>	Probability of recovery

12.57.1.2 ModelSIRMixing() [2/2]

```
template<typename TSeq >
ModelSIRMixing< TSeq >::ModelSIRMixing (
    const std::string & vname,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double recovery_rate,
    std::vector< double > contact_matrix ) [inline]
```

Constructs a [ModelSIRMixing](#) object.

Parameters

<i>vname</i>	The name of the ModelSIRMixing object.
<i>n</i>	The number of entities in the model.
<i>prevalence</i>	The initial prevalence of the disease in the model.
<i>contact_rate</i>	The contact rate between entities in the model.
<i>transmission_rate</i>	The transmission rate of the disease in the model.
<i>recovery_rate</i>	The recovery rate of the disease in the model.
<i>contact_matrix</i>	The contact matrix between entities in the model.

12.57.2 Member Function Documentation

12.57.2.1 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 >](#).

12.57.2.2 initial_states()

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

Set the initial states of the model.

Parameters

<i>proportions</i> ↪	Double vector with a single element:
—	<ul style="list-style-type: none"> The proportion of non-infected individuals who have recovered.

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

12.57.2.3 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

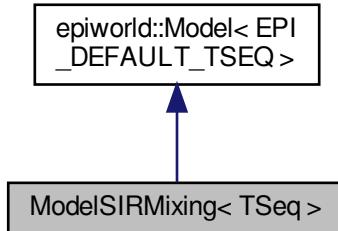
Reimplemented from [epiworld::Model< EPI_DEFAULT_TSEQ >](#).

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

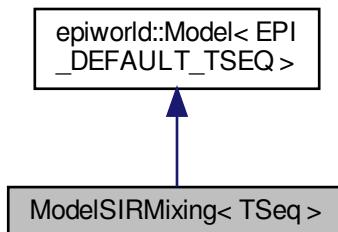
- `epiworld.hpp`

12.58 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

12.58.1 Constructor & Destructor Documentation

12.58.1.1 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

<code>model</code>	A reference to an existing <code>ModelSIRMixing</code> object.
<code>vname</code>	The name of the <code>ModelSIRMixing</code> object.
<code>n</code>	The number of entities in the model.
<code>prevalence</code>	The initial prevalence of the disease in the model.
<code>contact_rate</code>	The contact rate between entities in the model.
<code>transmission_rate</code>	The transmission rate of the disease in the model.
<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>model</code>	A <code>Model<TSeq></code> object where to set up the SIR.
<code>vname</code>	<code>std::string</code> Name of the virus
<code>prevalence</code>	Initial prevalence (proportion)
<code>contact_rate</code>	Average number of contacts (interactions) per step.
<code>transmission_rate</code>	Probability of transmission
<code>recovery_rate</code>	Probability of recovery

12.58.1.2 ModelSIRMixing() [2/2]

```
template<typename TSeq >
ModelSIRMixing< TSeq >::ModelSIRMixing (
    const std::string & vname,
    epiworld_fast_uint n,
    epiworld_double prevalence,
    epiworld_double contact_rate,
    epiworld_double transmission_rate,
    epiworld_double recovery_rate,
    std::vector< double > contact_matrix ) [inline]
```

Constructs a [ModelSIRMixing](#) object.

Parameters

<i>vname</i>	The name of the ModelSIRMixing object.
<i>n</i>	The number of entities in the model.
<i>prevalence</i>	The initial prevalence of the disease in the model.
<i>contact_rate</i>	The contact rate between entities in the model.
<i>transmission_rate</i>	The transmission rate of the disease in the model.
<i>recovery_rate</i>	The recovery rate of the disease in the model.
<i>contact_matrix</i>	The contact matrix between entities in the model.

12.58.2 Member Function Documentation

12.58.2.1 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>	<input type="checkbox"/>
-------------	--------------------------

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

12.58.2.2 initial_states()

```
template<typename TSeq >
ModelSIRMixing< TSeq > & ModelSIRMixing< TSeq >::initial_states (
```

```
std::vector< double > proportions_,
std::vector< int > queue_ = {} ) [inline], [virtual]
```

Set the initial states of the model.

Parameters

<i>proportions</i> ←	Double vector with a single element:
—	<ul style="list-style-type: none"> The proportion of non-infected individuals who have recovered.

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

12.58.2.3 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:

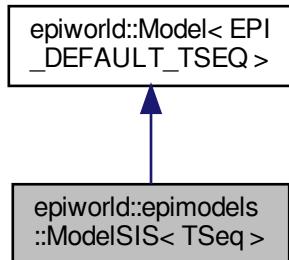
- include/epiworld/models/sirmixing.hpp

12.59 epiworld::epimodels::ModelSIS< TSeq > Class Template Reference

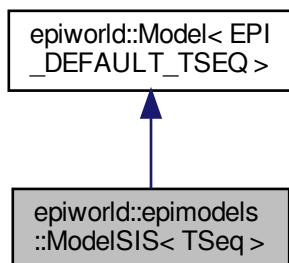
Template for a Susceptible-Infected-Susceptible (SIS) model.

```
#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

12.59.1 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:

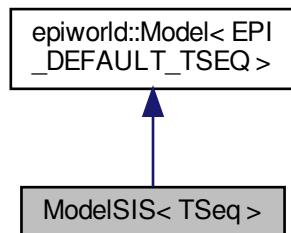
- epiworld.hpp

12.60 ModelSIS< TSeq > Class Template Reference

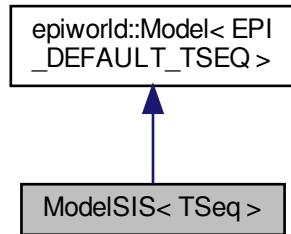
Template for a Susceptible-Infected-Susceptible (SIS) model.

```
#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

12.60.1 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:

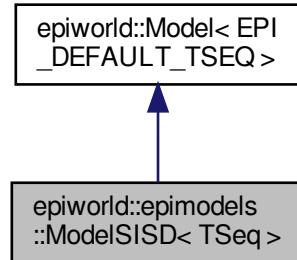
- include/epiworld/models/sis.hpp

12.61 epiworld::epimodels::ModelSISD< TSeq > Class Template Reference

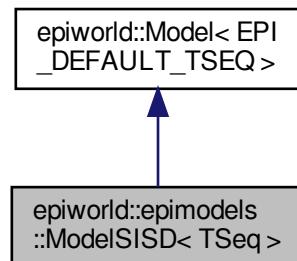
Template for a Susceptible-Infected-Susceptible-Deceased (SISD) model.

```
#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

12.61.1 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>initial_death</i>	epiworld_double Initial death_rate of the immune system

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

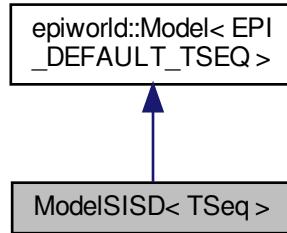
- epiworld.hpp

12.62 ModelSISD< TSeq > Class Template Reference

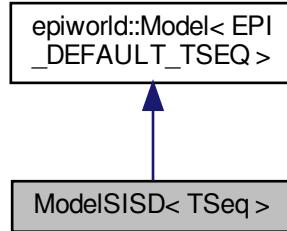
Template for a Susceptible-Infected-Susceptible-Deceased (SISD) model.

```
#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

12.62.1 Detailed Description

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

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

Parameters

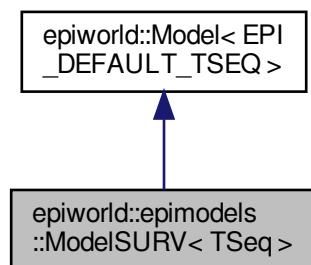
<i>vname</i>	std::string Name of the virus
<i>initial_prevalence</i>	epiworld_double Initial prevalence
<i>initial_efficacy</i>	epiworld_double Initial susceptibility_reduction of the immune system
<i>initial_recovery</i>	epiworld_double Initial recovery_rate rate of the immune system
<i>initial_death</i>	epiworld_double Initial death_rate of the immune system

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

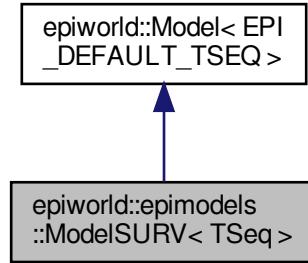
- include/epiworld/models/sisd.hpp

12.63 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 ($1 - P(\text{acquire the disease})$).</i>
<code>latent_period</code>	<i>Double. Shape parameter of a Gamma (<code>latent_period, 1</code>) distribution. This coincides with the expected number of latent days.</i>
<code>infect_period</code>	<i>Double. Shape parameter of a Gamma (<code>infected_period, 1</code>) 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**12.63.1 Member Function Documentation****12.63.1.1 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

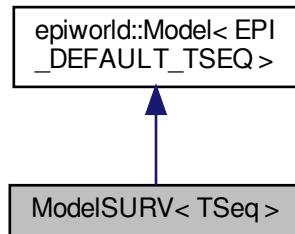
Reimplemented from `epiworld::Model< EPI_DEFAULT_TSEQ >`.

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

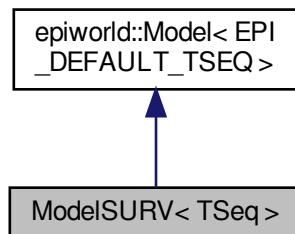
- `epiworld.hpp`

12.64 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 ($1 - P(\text{acquire the disease})$).</i>
latent_period	<i>Double. Shape parameter of a Gamma (<code>latent_period, 1</code>) distribution. This coincides with the expected number of latent days.</i>

Parameters

infect_period	<i>Double.</i> Shape parameter of a <i>Gamma</i> (<i>infected_period, 1</i>) distribution. This coincides with the expected number of infectious days.
prob_symptoms	<i>Double.</i> Probability of generating symptoms.
prop_vaccinated	<i>Double.</i> Probability of vaccination. Coincides with the initial prevalence of vaccinated individuals.
prop_vax_redux_transm	<i>Double.</i> Factor by which the vaccine reduces transmissibility.
prop_vax_redux_infect	<i>Double.</i> Factor by which the vaccine reduces the chances of becoming infected.
surveillance_prob	<i>Double.</i> Probability of testing an agent.
prob_transmission	<i>Double.</i> Raw transmission probability.
prob_death	<i>Double.</i> Raw probability of death for symptomatic individuals.
prob_noreinfect	<i>Double.</i> Probability of no re-infection.

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**12.64.1 Member Function Documentation**

12.64.1.1 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

12.65 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

12.66 epiworld::PersonTools< TSeq > Class Template Reference

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

- epiworld.hpp

12.67 PersonTools< TSeq > Class Template Reference

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

- include/epiworld/config.hpp

12.68 epiworld::Progress Class Reference

A simple progress bar.

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

Public Member Functions

- **Progress** (int n_, int width_)
- void **start** ()
- void **next** ()

12.68.1 Detailed Description

A simple progress bar.

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

- epiworld.hpp

12.69 Progress Class Reference

A simple progress bar.

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

Public Member Functions

- **Progress** (int n_, int width_)
- void **start** ()
- void **next** ()

12.69.1 Detailed Description

A simple progress bar.

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

- include/epiworld/progress.hpp

12.70 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 >](#)

12.70.1 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

12.71 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 >**

12.71.1 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

12.72 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

12.73 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

12.74 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

12.75 TestResults Struct Reference

Public Attributes

- epimodels::ModelMeaslesMixingRiskQuarantine **model**
- std::vector< epiworld_double > **transitions**
- std::vector< epiworld_double > **R0s**
- std::vector< std::vector< int > > **final_distribution**
- std::vector< double > **sizes**

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

- tests/19c-measles-mixing-risk-quarantine-durations.cpp

12.76 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)

12.76.1 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

12.77 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)

12.77.1 Detailed Description

```
template<typename TSeq>
class Tool< TSeq >
```

[Tools](#) for defending the agent against the virus.

Template Parameters

TSeq	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

12.78 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

12.78.1 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

12.79 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

12.79.1 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

12.80 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 >**

12.80.1 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

12.81 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 >**

12.81.1 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

12.82 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** >

12.82.1 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

12.83 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** >

12.83.1 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

12.84 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

<i>x</i>	<i>A vector of length ncol () (if vector), otherwise a epiworld_double.</i>
<i>j</i>	<i>Index of the data point, from 0 to ncol () - 1.</i>

- [void add \(std::vector< epiworld_double > x\)](#)
- [void add \(epiworld_fast_uint j, epiworld_double x\)](#)

Access data

Parameters

<i>i</i>	<i>Row (0 through ndays - 1.)</i>
<i>j</i>	<i>Column (0 through 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 >**

12.84.1 Detailed Description

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

Personalized data by the user.

Template Parameters

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

12.84.2 Constructor & Destructor Documentation**12.84.2.1 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*

12.85 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

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 () - 1</code> .

- [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 - 1</code> .)
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 >](#)

12.85.1 Detailed Description

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

Personalized data by the user.

Template Parameters

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

12.85.2 Constructor & Destructor Documentation**12.85.2.1 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

12.86 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

12.86.1 Detailed Description

```
template<typename T>
struct epiworld::vecHasher< T >
```

Vector hasher.

Template Parameters

T	
-----	--

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

- epiworld.hpp

12.87 vecHasher< T > Struct Template Reference

Vector hasher.

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

Public Member Functions

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

12.87.1 Detailed Description

```
template<typename T>
struct vecHasher< T >
```

Vector hasher.

Template Parameters

T	
-----	--

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

- include/epiworld/misc.hpp

12.88 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	<i>After the virus/tool is added to the agent.</i>
end	<i>After the virus/tool is removed.</i>
removed	<i>After the agent (Agent) is removed.</i>

- 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)

12.88.1 Detailed Description

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

[Virus](#).

Template Parameters

TSeq	
------	--

Raw transmisibility of a virus should be a function of its genetic sequence. Nonetheless, transmisibility can be reduced as a result of having one or more tools to fight the virus. Because of this, transmisibility should be a function of the agent.

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

- epiworld.hpp

12.89 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_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	<i>After the virus/tool is added to the agent.</i>
end	<i>After the virus/tool is removed.</i>
removed	<i>After the agent (Agent) is removed.</i>

- 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)

12.89.1 Detailed Description

```
template<typename TSeq>
class Virus< TSeq >
```

[Virus](#).

Template Parameters

TSeq	
------	--

Raw transmisibility of a virus should be a function of its genetic sequence. Nonetheless, transmisibility can be reduced as a result of having one or more tools to fight the virus. Because of this, transmisibility 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

12.90 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 >**

12.90.1 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

12.91 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 >**

12.91.1 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

12.92 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 >

12.92.1 Detailed Description

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

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

Template Parameters

TSeq	
------	--

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

- epiworld.hpp

12.93 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 >

12.93.1 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

12.94 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

12.95 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

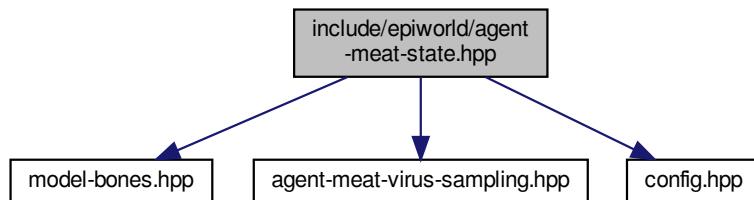
Chapter 13

File Documentation

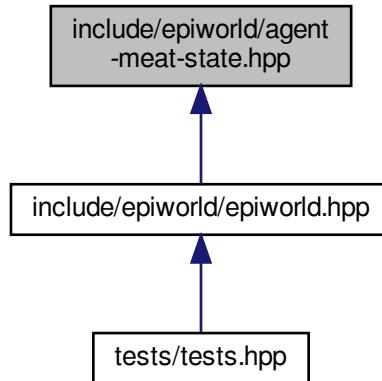
13.1 include/epiworld/agent-meat-state.hpp File Reference

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)

13.1.1 Detailed Description

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

Author

George G. Vega Yon (g.vegaton en gmail)

Version

0.1

Date

2022-06-15

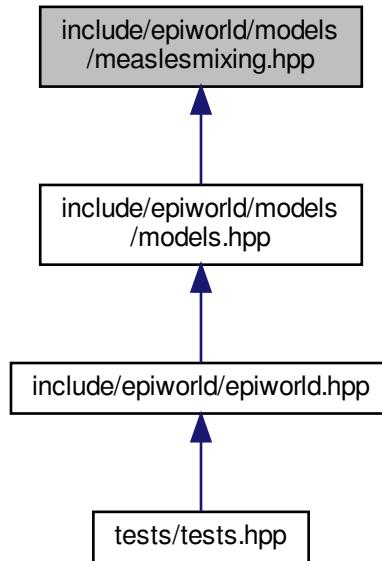
Copyright

Copyright (c) 2022

13.2 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)

13.2.1 Detailed Description

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

13.2.2 Macro Definition Documentation

13.2.2.1 GET_MODEL

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

Value:

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

13.2.2.2 SAMPLE_FROM_PROBS

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

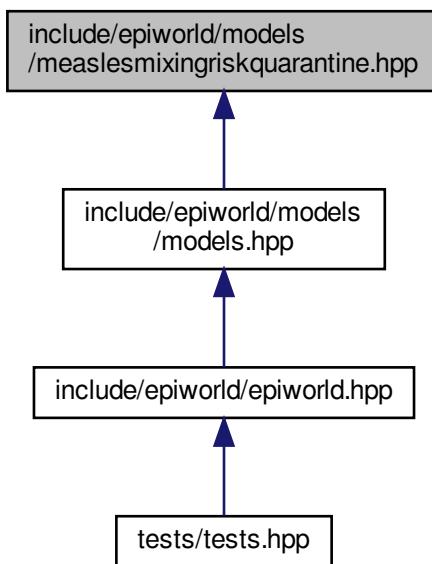
Value:

```
size_t ans; \
epiworld_double p_total = m->runeif(); \
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]; \
}
```

13.3 include/epiworld/models/measlesmixingriskquarantine.hpp File Reference

Template for a Measles model with population mixing and risk-based quarantine.

This graph shows which files directly or indirectly include this file:



Classes

- class [ModelMeaslesMixingRiskQuarantine< TSeq >](#)
Measles model with population mixing and risk-based quarantine strategies.

Macros

- `#define COL_MAJOR_POS(i, j, n) (j * n + i)`
- `#define GET_MODEL(model, output)`
- `#define SAMPLE_FROM_PROBS(n, ans)`
Macro to sample from a list of probabilities.

13.3.1 Detailed Description

Template for a Measles model with population mixing and risk-based quarantine.

13.3.2 Macro Definition Documentation

13.3.2.1 GET_MODEL

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

Value:

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

13.3.2.2 SAMPLE_FROM_PROBS

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

Value:

```
int ans; \
epiworld_double p_total = m->runif(); \
for (ans = 0; ans < static_cast<int>(n); ans++) \
{ \
    if (p_total < m->array_double_tmp[ans]) \
        break; \
    m->array_double_tmp[ans + 1] += m->array_double_tmp[ans]; \
};
```

Macro to sample from a list of probabilities.

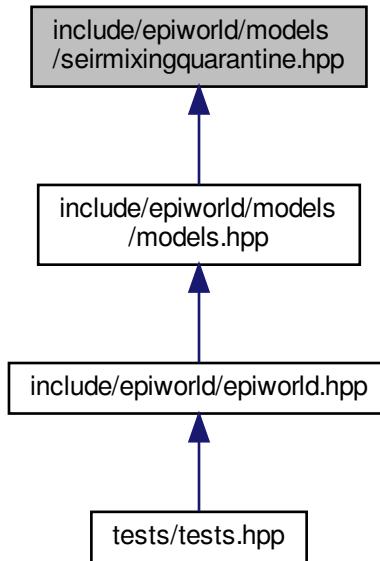
Returns

The index of the sampled probability; and the total length if none is found, returns n.

13.4 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)`

13.4.1 Detailed Description

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

13.4.2 Macro Definition Documentation

13.4.2.1 GET_MODEL

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

Value:

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

13.4.2.2 SAMPLE_FROM_PROBS

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

Value:

```
size_t ans; \
epiworld_double p_total = m->runif(); \
for (ans = 0; 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_contact
 ContactTracing, 54
 epiworld::ContactTracing, 57

add_globalevent
 epiworld::Model< TSeq >, 93
 Model< TSeq >, 109

AdjList, 39
 AdjList, 39
 epiworld::AdjList, 41
 read_edgelist, 40

Agent< TSeq >, 42
 default_rm_entity, 46
 operator(), 44
 swap_neighbors, 46

AgentsSample
 AgentsSample< TSeq >, 51
 epiworld::AgentsSample< TSeq >, 53

AgentsSample< TSeq >, 50
 AgentsSample, 51

clone_ptr
 epiworld::epimodels::ModelMeaslesMixing< TSeq >, 126
 epiworld::epimodels::ModelMeaslesMixingRiskQuarantine< TSeq >, 142
 epiworld::epimodels::ModelMeaslesSchool< TSeq >, 156
 epiworld::epimodels::ModelSEIRCONN< TSeq >, 169
 epiworld::epimodels::ModelSEIRDCONN< TSeq >, 182
 epiworld::epimodels::ModelSEIRMixing< TSeq >, 189
 epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >, 199
 epiworld::epimodels::ModelSIRCONN< TSeq >, 215
 epiworld::epimodels::ModelSIRDCONN< TSeq >, 228
 epiworld::epimodels::ModelSIRLogit< TSeq >, 234
 epiworld::epimodels::ModelSIRMixing< TSeq >, 240
 epiworld::Model< TSeq >, 93
 Model< TSeq >, 109
 ModelMeaslesMixing< TSeq >, 134
 ModelMeaslesMixingRiskQuarantine< TSeq >, 150
 ModelMeaslesSchool< TSeq >, 160
 ModelSEIRCONN< TSeq >, 172

ModelSEIRDCONN< TSeq >, 185
ModelSEIRMixing< TSeq >, 192
ModelSEIRMixingQuarantine< TSeq >, 207
ModelSIRCONN< TSeq >, 219
ModelSIRDCONN< TSeq >, 230
ModelSIRLogit< TSeq >, 237
ModelSIRMixing< TSeq >, 244

ContactTracing, 53
 add_contact, 54
 ContactTracing, 54
 epiworld::ContactTracing, 57
 get_contact, 55
 get_n_contacts, 55
 print, 56
 reset, 56

DataBase< TSeq >, 59
 get_generation_time, 62
 get_reproductive_number, 62
 get_transition_probability, 62
 get_transmissions, 63
 operator==, 63, 64
 record_virus, 64

default_rm_entity
 Agent< TSeq >, 46
 Entity< TSeq >, 74
 epiworld::Agent< TSeq >, 50
 epiworld::Entity< TSeq >, 75

draw
 epiworld::Model< TSeq >, 94
 Model< TSeq >, 109

Entities< TSeq >, 69
Entities_const< TSeq >, 71

Entity
 Entity< TSeq >, 73
 epiworld::Entity< TSeq >, 75

Entity< TSeq >, 73
 default_rm_entity, 74
 Entity, 73

epiworld::AdjList, 40
 AdjList, 41
 read_edgelist, 41

epiworld::Agent< TSeq >, 47
 default_rm_entity, 50
 operator(), 49
 swap_neighbors, 49

epiworld::AgentsSample< TSeq >, 52
 AgentsSample, 53

epiworld::ContactTracing, 56

add_contact, 57
 ContactTracing, 57
 get_contact, 58
 get_n_contacts, 58
 print, 59
 reset, 59
 epiworld::DataBase< TSeq >, 65
 get_generation_time, 67
 get_reproductive_number, 67
 get_transition_probability, 68
 get_transmissions, 68
 operator==, 69
 record_virus, 69
 epiworld::Entities< TSeq >, 70
 epiworld::Entities_const< TSeq >, 72
 epiworld::Entity< TSeq >, 74
 default_rm_entity, 75
 Entity, 75
 epiworld::epimodels::ModelDiffNet< TSeq >, 117
 epiworld::epimodels::ModelMeaslesMixing< TSeq >, 120
 clone_ptr, 126
 get_agent_quarantine_triggered, 126
 get_contact_matrix, 126
 get_isolation_willingness, 127
 get_quarantine_willingness, 127
 initial_states, 127
 ModelMeaslesMixing, 123, 125
 run, 128
 set_contact_matrix, 128
 epiworld::epimodels::ModelMeaslesMixingRiskQuarantine< TSeq >, 137
 clone_ptr, 142
 get_contact_matrix, 142
 get_days_quarantine_triggered, 142
 get_isolation_willingness, 143
 get_quarantine_risk_level, 143
 get_quarantine_willingness, 143
 initial_states, 143
 ModelMeaslesMixingRiskQuarantine, 140, 141
 run, 144
 set_contact_matrix, 144
 epiworld::epimodels::ModelMeaslesSchool< TSeq >, 153
 clone_ptr, 156
 ModelMeaslesSchool, 155
 quarantine_agents, 156
 reset, 156
 epiworld::epimodels::ModelSEIR< TSeq >, 162
 initial_states, 163
 update_exposed_seir, 164
 update_infected_seir, 164
 epiworld::epimodels::ModelSEIRCONN< TSeq >, 167
 clone_ptr, 169
 initial_states, 169
 ModelSEIRCONN, 168
 reset, 170
 epiworld::epimodels::ModelSEIRD< TSeq >, 173
 ModelSEIRD, 175, 176
 update_exposed_seir, 176
 epiworld::epimodels::ModelSEIRDCONN< TSeq >, 180
 clone_ptr, 182
 initial_states, 182
 ModelSEIRDCONN, 181
 reset, 182
 epiworld::epimodels::ModelSEIRMixing< TSeq >, 186
 clone_ptr, 189
 initial_states, 189
 ModelSEIRMixing, 187, 188
 reset, 189
 epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >, 194
 clone_ptr, 199
 get_agent_quarantine_triggered, 199
 get_contact_matrix, 199
 get_isolation_willingness, 200
 get_quarantine_willingness, 200
 initial_states, 200
 ModelSEIRMixingQuarantine, 196, 198
 run, 201
 set_contact_matrix, 201
 epiworld::epimodels::ModelSIR< TSeq >, 210
 initial_states, 211
 epiworld::epimodels::ModelSIRCONN< TSeq >, 214
 clone_ptr, 215
 get_n_infected, 216
 initial_states, 216
 ModelSIRCONN, 215
 reset, 216
 epiworld::epimodels::ModelSIRD< TSeq >, 220
 initial_states, 223
 ModelSIRD, 221
 epiworld::epimodels::ModelSIRDCONN< TSeq >, 226
 clone_ptr, 228
 ModelSIRDCONN, 227
 reset, 228
 epiworld::epimodels::ModelSIRLogit< TSeq >, 231
 clone_ptr, 234
 ModelSIRLogit, 233
 reset, 234
 epiworld::epimodels::ModelSIRMixing< TSeq >, 238
 clone_ptr, 240
 initial_states, 241
 ModelSIRMixing, 239, 240
 reset, 241
 epiworld::epimodels::ModelSIS< TSeq >, 245
 epiworld::epimodels::ModelISID< TSeq >, 249
 epiworld::epimodels::ModelSURV< TSeq >, 252
 reset, 254
 epiworld::Event< TSeq >, 76
 Event, 77
 epiworld::GlobalEvent< TSeq >, 79
 GlobalEvent, 80
 epiworld::LFMCMC< TData >, 82
 epiworld::Model< TSeq >, 85

add_globalevent, 93
clone_ptr, 93
draw, 94
events_add, 94
events_run, 95
initial_states_fun, 98
load_agents_entities_ties, 95
rbinomd, 98
reset, 96
rexpd, 98
rgammaad, 98
rgeomd, 99
rlognormald, 99
rnbinomd, 99
rnormd, 99
rpoissd, 99
run_multiple, 96
runifd, 100
set_agents_data, 96
set_name, 97
time_elapsed, 100
write_data, 97
epiworld::ModelDiagram, 117
epiworld::PersonTools< TSeq >, 257
epiworld::Progress, 258
epiworld::Queue< TSeq >, 259
epiworld::sampler, 31
 make_sample_virus_neighbors, 31
 make_update_susceptible, 32
 sample_virus_single, 32
epiworld::SAMPLETYPE, 261
epiworld::Tool< TSeq >, 261
epiworld::ToolFunctions< TSeq >, 264
epiworld::Tools< TSeq >, 266
epiworld::Tools_const< TSeq >, 267
epiworld::UserData< TSeq >, 269
 UserData, 270
epiworld::vechasher< T >, 272
epiworld::Virus< TSeq >, 273
epiworld::Viruses< TSeq >, 278
epiworld::Viruses_const< TSeq >, 279
epiworld::VirusFunctions< TSeq >, 281
Event
 epiworld::Event< TSeq >, 77
 Event< TSeq >, 79
Event< TSeq >, 77
 Event, 79
events_add
 epiworld::Model< TSeq >, 94
 Model< TSeq >, 110
events_run
 epiworld::Model< TSeq >, 95
 Model< TSeq >, 110
get_agent_quarantine_triggered
 epiworld::epimodels::ModelMeaslesMixing< TSeq >, 126
 epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >, 199
ModelMeaslesMixing< TSeq >, 135
ModelSEIRMixingQuarantine< TSeq >, 207
get_contact
 ContactTracing, 55
 epiworld::ContactTracing, 58
get_contact_matrix
 epiworld::epimodels::ModelMeaslesMixing< TSeq >, 126
 epiworld::epimodels::ModelMeaslesMixingRiskQuarantine< TSeq >, 142
 epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >, 199
 ModelMeaslesMixing< TSeq >, 135
 ModelMeaslesMixingRiskQuarantine< TSeq >, 150
 ModelSEIRMixingQuarantine< TSeq >, 207
get_days_quarantine_triggered
 epiworld::epimodels::ModelMeaslesMixingRiskQuarantine< TSeq >, 142
 ModelMeaslesMixingRiskQuarantine< TSeq >, 150
get_generation_time
 DataBase< TSeq >, 62
 epiworld::DataBase< TSeq >, 67
get_isolation_willingness
 epiworld::epimodels::ModelMeaslesMixing< TSeq >, 127
 epiworld::epimodels::ModelMeaslesMixingRiskQuarantine< TSeq >, 143
 epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >, 200
 ModelMeaslesMixing< TSeq >, 135
 ModelMeaslesMixingRiskQuarantine< TSeq >, 150
 ModelSEIRMixingQuarantine< TSeq >, 208
GET_MODEL
 measlesmixing.hpp, 285
 measlesmixingriskquarantine.hpp, 287
 seirmixingquarantine.hpp, 288
get_n_contacts
 ContactTracing, 55
 epiworld::ContactTracing, 58
get_n_infected
 epiworld::epimodels::ModelSIRCONN< TSeq >, 216
 ModelSIRCONN< TSeq >, 219
get_quarantine_risk_level
 epiworld::epimodels::ModelMeaslesMixingRiskQuarantine< TSeq >, 143
 ModelMeaslesMixingRiskQuarantine< TSeq >, 151
get_quarantine_willingness
 epiworld::epimodels::ModelMeaslesMixing< TSeq >, 127
 epiworld::epimodels::ModelMeaslesMixingRiskQuarantine< TSeq >, 143
 epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >, 200

ModelMeaslesMixing< TSeq >, 135
 ModelMeaslesMixingRiskQuarantine< TSeq >, 151
 ModelSEIRMixingQuarantine< TSeq >, 208
 get_reproductive_number
 DataBase< TSeq >, 62
 epiworld::DataBase< TSeq >, 67
 get_transition_probability
 DataBase< TSeq >, 62
 epiworld::DataBase< TSeq >, 68
 get_transmissions
 DataBase< TSeq >, 63
 epiworld::DataBase< TSeq >, 68
 GlobalEvent
 epiworld::GlobalEvent< TSeq >, 80
 GlobalEvent< TSeq >, 81
 GlobalEvent< TSeq >, 81
 GlobalEvent, 81

 include/epiworld/agent-meat-state.hpp, 283
 include/epiworld/models/measlesmixing.hpp, 285
 include/epiworld/models/measlesmixingriskquarantine.hpp, 286
 include/epiworld/models/seirmixingquarantine.hpp, 288
 initial_states
 epiworld::epimodels::ModelMeaslesMixing< TSeq >, 127
 epiworld::epimodels::ModelMeaslesMixingRiskQuarantine< TSeq >, 143
 epiworld::epimodels::ModelSEIR< TSeq >, 163
 epiworld::epimodels::ModelSEIRCONN< TSeq >, 169
 epiworld::epimodels::ModelSEIRDCONN< TSeq >, 182
 epiworld::epimodels::ModelSEIRMixing< TSeq >, 189
 epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >, 200
 epiworld::epimodels::ModelSIR< TSeq >, 211
 epiworld::epimodels::ModelSIRCONN< TSeq >, 216
 epiworld::epimodels::ModelSIRD< TSeq >, 223
 epiworld::epimodels::ModelSIRMixing< TSeq >, 241
 ModelMeaslesMixing< TSeq >, 136
 ModelMeaslesMixingRiskQuarantine< TSeq >, 151
 ModelSEIR< TSeq >, 166
 ModelSEIRCONN< TSeq >, 172
 ModelSEIRDCONN< TSeq >, 185
 ModelSEIRMixing< TSeq >, 193
 ModelSEIRMixingQuarantine< TSeq >, 208
 ModelSIR< TSeq >, 213
 ModelSIRCONN< TSeq >, 219
 ModelSIRD< TSeq >, 225
 ModelSIRMixing< TSeq >, 244
 initial_states_fun
 epiworld::Model< TSeq >, 98
 Model< TSeq >, 114

 LFMCMC< TData >, 83
 load_agents_entities_ties
 epiworld::Model< TSeq >, 95
 Model< TSeq >, 112
 make_sample_virus_neighbors
 epiworld::sampler, 31
 sampler, 34
 make_update_susceptible
 epiworld::sampler, 32
 sampler, 35
 measlesmixing.hpp
 GET_MODEL, 285
 SAMPLE_FROM_PROBS, 286
 measlesmixingriskquarantine.hpp
 GET_MODEL, 287
 SAMPLE_FROM_PROBS, 287
 Model< TSeq >, 100
 add_globalevent, 109
 clone_ptr, 109
 draw, 109
 events_add, 110
 events_run, 110
 initial_states_fun, 114
 load_agents_entities_ties, 112
 rbinomd, 114
 reset, 112
 rgammaid, 115
 rgeomd, 115
 rlognormald, 115
 rnbinomd, 115
 rnormd, 115
 rpoissd, 116
 run_multiple, 112
 runifd, 116
 set_agents_data, 113
 set_name, 113
 time_elapsed, 116
 write_data, 114
 ModelDiagram, 117
 ModelDiffNet< TSeq >, 119
 ModelMeaslesMixing
 epiworld::epimodels::ModelMeaslesMixing< TSeq >, 123, 125
 ModelMeaslesMixing< TSeq >, 132, 133
 ModelMeaslesMixing< TSeq >, 129
 clone_ptr, 134
 get_agent_quarantine_triggered, 135
 get_contact_matrix, 135
 get_isolation_willingness, 135
 get_quarantine_willingness, 135
 initial_states, 136
 ModelMeaslesMixing, 132, 133
 run, 136
 set_contact_matrix, 137
 ModelMeaslesMixingRiskQuarantine
 epiworld::epimodels::ModelMeaslesMixingRiskQuarantine< TSeq >, 140, 141

ModelMeaslesMixingRiskQuarantine< TSeq >, 147, 149
ModelMeaslesMixingRiskQuarantine< TSeq >, 145
clone_ptr, 150
get_contact_matrix, 150
get_days_quarantine_triggered, 150
get_isolation_willingness, 150
get_quarantine_risk_level, 151
get_quarantine_willingness, 151
initial_states, 151
ModelMeaslesMixingRiskQuarantine, 147, 149
run, 152
set_contact_matrix, 152
ModelMeaslesSchool
epiworld::epimodels::ModelMeaslesSchool< TSeq >, 155
ModelMeaslesSchool< TSeq >, 159
ModelMeaslesSchool< TSeq >, 157
clone_ptr, 160
ModelMeaslesSchool, 159
quarantine_agents, 161
reset, 161
ModelSEIR< TSeq >, 165
initial_states, 166
update_exposed_seir, 167
update_infected_seir, 167
ModelSEIRCONN
epiworld::epimodels::ModelSEIRCONN< TSeq >, 168
ModelSEIRCONN< TSeq >, 171
ModelSEIRCONN< TSeq >, 170
clone_ptr, 172
initial_states, 172
ModelSEIRCONN, 171
reset, 173
ModelSEIRD
epiworld::epimodels::ModelSEIRD< TSeq >, 175, 176
ModelSEIRD< TSeq >, 178, 179
ModelSEIRD< TSeq >, 177
ModelSEIRD, 178, 179
update_exposed_seir, 179
ModelSEIRDCONN
epiworld::epimodels::ModelSEIRDCONN< TSeq >, 181
ModelSEIRDCONN< TSeq >, 184
ModelSEIRDCONN< TSeq >, 183
clone_ptr, 185
initial_states, 185
ModelSEIRDCONN, 184
reset, 186
ModelSEIRMixing
epiworld::epimodels::ModelSEIRMixing< TSeq >, 187, 188
ModelSEIRMixing< TSeq >, 191, 192
ModelSEIRMixing< TSeq >, 190
clone_ptr, 192
initial_states, 193
ModelSEIRMixing, 191, 192
reset, 193
ModelSEIRMixingQuarantine
epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >, 196, 198
ModelSEIRMixingQuarantine< TSeq >, 204, 206
ModelSEIRMixingQuarantine< TSeq >, 202
clone_ptr, 207
get_agent_quarantine_triggered, 207
get_contact_matrix, 207
get_isolation_willingness, 208
get_quarantine_willingness, 208
initial_states, 208
ModelSEIRMixingQuarantine, 204, 206
run, 209
set_contact_matrix, 209
ModelSIR< TSeq >, 211
initial_states, 213
ModelSIRCONN
epiworld::epimodels::ModelSIRCONN< TSeq >, 215
ModelSIRCONN< TSeq >, 218
ModelSIRCONN< TSeq >, 217
clone_ptr, 219
get_n_infected, 219
initial_states, 219
ModelSIRCONN, 218
reset, 220
ModelSIRD
epiworld::epimodels::ModelSIRD< TSeq >, 221
ModelSIRD< TSeq >, 225
ModelSIRD< TSeq >, 223
initial_states, 225
ModelSIRD, 225
ModelSIRDCONN
epiworld::epimodels::ModelSIRDCONN< TSeq >, 227
ModelSIRDCONN< TSeq >, 230
ModelSIRDCONN< TSeq >, 229
clone_ptr, 230
ModelSIRDCONN, 230
reset, 231
ModelSIRLogit
epiworld::epimodels::ModelSIRLogit< TSeq >, 233
ModelSIRLogit< TSeq >, 236
ModelSIRLogit< TSeq >, 235
clone_ptr, 237
ModelSIRLogit, 236
reset, 237
ModelSIRMixing
epiworld::epimodels::ModelSIRMixing< TSeq >, 239, 240
ModelSIRMixing< TSeq >, 243, 244
ModelSIRMixing< TSeq >, 242
clone_ptr, 244
initial_states, 244
ModelSIRMixing, 243, 244

reset, 245
 ModelSIS< TSeq >, 248
 ModelSISD< TSeq >, 251
 ModelSURV< TSeq >, 255
 reset, 256

 Network< Nettype, Nodetype, Edgetype >, 257

 operator()
 Agent< TSeq >, 44
 epiworld::Agent< TSeq >, 49
 operator==
 DataBase< TSeq >, 63, 64
 epiworld::DataBase< TSeq >, 69

 PersonTools< TSeq >, 258
 print
 ContactTracing, 56
 epiworld::ContactTracing, 59
 Progress, 258

 quarantine_agents
 epiworld::epimodels::ModelMeaslesSchool< TSeq >, 156
 ModelMeaslesSchool< TSeq >, 161
 Queue< TSeq >, 259

 RandGraph, 260
 rbinomd
 epiworld::Model< TSeq >, 98
 Model< TSeq >, 114
 read_edgelist
 AdjList, 40
 epiworld::AdjList, 41
 record_virus
 DataBase< TSeq >, 64
 epiworld::DataBase< TSeq >, 69
 reset
 ContactTracing, 56
 epiworld::ContactTracing, 59
 epiworld::epimodels::ModelMeaslesSchool< TSeq >, 156
 epiworld::epimodels::ModelSEIRCONN< TSeq >, 170
 epiworld::epimodels::ModelSEIRDCCONN< TSeq >, 182
 epiworld::epimodels::ModelSEIRMixing< TSeq >, 189
 epiworld::epimodels::ModelSIRCONN< TSeq >, 216
 epiworld::epimodels::ModelSIRDCCONN< TSeq >, 228
 epiworld::epimodels::ModelSIRLogit< TSeq >, 234
 epiworld::epimodels::ModelSIRMixing< TSeq >, 241
 epiworld::epimodels::ModelSURV< TSeq >, 254
 epiworld::Model< TSeq >, 96
 Model< TSeq >, 112

 ModelMeaslesSchool< TSeq >, 161
 ModelSEIRCONN< TSeq >, 173
 ModelSEIRDCCONN< TSeq >, 186
 ModelSEIRMixing< TSeq >, 193
 ModelSIRCONN< TSeq >, 220
 ModelSIRDCCONN< TSeq >, 231
 ModelSIRLogit< TSeq >, 237
 ModelSIRMixing< TSeq >, 245
 ModelSURV< TSeq >, 256
 rexpd
 epiworld::Model< TSeq >, 98
 Model< TSeq >, 114
 rgammaad
 epiworld::Model< TSeq >, 98
 Model< TSeq >, 115
 rgeomd
 epiworld::Model< TSeq >, 99
 Model< TSeq >, 115
 rlognormad
 epiworld::Model< TSeq >, 99
 Model< TSeq >, 115
 rnbinomd
 epiworld::Model< TSeq >, 99
 Model< TSeq >, 115
 rnormd
 epiworld::Model< TSeq >, 99
 Model< TSeq >, 115
 rpoissd
 epiworld::Model< TSeq >, 99
 Model< TSeq >, 116
 run
 epiworld::epimodels::ModelMeaslesMixing< TSeq >, 128
 epiworld::epimodels::ModelMeaslesMixingRiskQuarantine< TSeq >, 144
 epiworld::epimodels::ModelSEIRMixingQuarantine< TSeq >, 201
 ModelMeaslesMixing< TSeq >, 136
 ModelMeaslesMixingRiskQuarantine< TSeq >, 152
 ModelSEIRMixingQuarantine< TSeq >, 209
 run_multiple
 epiworld::Model< TSeq >, 96
 Model< TSeq >, 112
 runifd
 epiworld::Model< TSeq >, 100
 Model< TSeq >, 116
 SAMPLE_FROM_PROBS
 measlesmixing.hpp, 286
 measlesmixingriskquarantine.hpp, 287
 seirmixingquarantine.hpp, 289
 sample_virus_single
 epiworld::sampler, 32
 sampler, 35
 sampler, 34
 make_sample_virus_neighbors, 34
 make_update_susceptible, 35
 sample_virus_single, 35

SAMPLETYPE, 261
seirmixingquarantine.hpp
 GET_MODEL, 288
 SAMPLE_FROM_PROBS, 289
set_agents_data
 epiworld::Model< TSeq >, 96
 Model< TSeq >, 113
set_contact_matrix
 epiworld::epimodels::ModelMeaslesMixing< TSeq
 >, 128
 epiworld::epimodels::ModelMeaslesMixingRiskQuarantine<
 TSeq >, 144
 epiworld::epimodels::ModelSEIRMixingQuarantine<
 TSeq >, 201
 ModelMeaslesMixing< TSeq >, 137
 ModelMeaslesMixingRiskQuarantine< TSeq >, 152
 ModelSEIRMixingQuarantine< TSeq >, 209
set_name
 epiworld::Model< TSeq >, 97
 Model< TSeq >, 113
swap_neighbors
 Agent< TSeq >, 46
 epiworld::Agent< TSeq >, 49
TestResults, 261
time_elapsed
 epiworld::Model< TSeq >, 100
 Model< TSeq >, 116
Tool< TSeq >, 263
ToolFunctions< TSeq >, 265
Tools< TSeq >, 266
Tools_const< TSeq >, 268
update_exposed_seir
 epiworld::epimodels::ModelSEIR< TSeq >, 164
 epiworld::epimodels::ModelSEIRD< TSeq >, 176
 ModelSEIR< TSeq >, 167
 ModelSEIRD< TSeq >, 179
update_infected_seir
 epiworld::epimodels::ModelSEIR< TSeq >, 164
 ModelSEIR< TSeq >, 167
UserData
 epiworld::UserData< TSeq >, 270
 UserData< TSeq >, 272
UserData< TSeq >, 270
 UserData, 272
vecHasher< T >, 273
Virus< TSeq >, 275
Viruses< TSeq >, 278
Viruses_const< TSeq >, 280
VirusFunctions< TSeq >, 281
write_data
 epiworld::Model< TSeq >, 97
 Model< TSeq >, 114