```
TRANSLATING MODEL 'BLCA_unconstrained_Givens_fixC_noZ' FROM Stan CODE TO C++
CODE NOW.
successful in parsing the Stan model 'BLCA_unconstrained_Givens_fixC_noZ'.
DIAGNOSTIC(S) FROM PARSER:
Info: integer division implicitly rounds to integer. Found int division: Q - 1 * Q + 2 / 2
 Positive values rounded down, negative values rounded up or down in platform-dependent way.

COMPILING THE C++ CODE FOR MODEL 'BLCA_unconstrained_Givens_fixC_noZ' NOW.
OS: x86_64, linux-gnu; rstan: 2.19.2; Rcpp: 1.0.2; inline: 0.3.15
 >> setting environment variables:
PKG_LIBS =  -L'/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/lib/' -
lStanHeaders
PKG_CPPFLAGS =   -I"/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/Rcpp/include/"  -I"/
home/changy/R/x86_64-redhat-linux-gnu-library/3.6/RcppEigen/include/"  -I"/home/changy/R/
x86_64-redhat-linux-gnu-library/3.6/RcppEigen/include/unsupported"  -I"/home/changy/R/
x86_64-redhat-linux-gnu-library/3.6/BH/include" -I"/home/changy/R/x86_64-redhat-linux-gnu-
library/3.6/StanHeaders/include/src/"  -I"/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/
StanHeaders/include/"  -I"/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/rstan/include" -
DEIGEN_NO_DEBUG  -DBOOST_DISABLE_ASSERTS
 >> Program source :

  1 :
  2 : // includes from the plugin
  3 : // [[Rcpp::plugins(cpp14)]]
  4 :
  5 : // user includes
  6 : #define STAN__SERVICES__COMMAND_HPP// Code generated by Stan version 2.19.1
  7 :
  8 : #include <stan/model/model_header.hpp>
  9 :
 10 : namespace model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace {
 11 :
 12 : using std::istream;
 13 : using std::string;
 14 : using std::stringstream;
 15 : using std::vector;
 16 : using stan::io::dump;
 17 : using stan::math::lgamma;
 18 : using stan::model::prob_grad;
 19 : using namespace stan::math;
 20 :
 21 : static int current_statement_begin__;
 22 :
 23 : stan::io::program_reader prog_reader__() {
 24 :     stan::io::program_reader reader;
 25 :     reader.add_event(0, 0, "start",
"model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ");
```

```
  26 :     reader.add_event(164, 162, "end",
"model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ");
  27 :     return reader;
  28 : }
  29 :
  30 : class model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ : public prob_grad {
  31 : private:
  32 :         int TT;
  33 :         int Q;
  34 :         std::vector<int> C;
  35 :         int V;
  36 :         matrix_d X;
  37 :         double delta_w;
  38 :         double prior_mean_mu_free;
  39 :         double prior_precision_mu_free;
  40 :         double prior_mean_log_sigma_free_sq;
  41 :         double prior_sd_log_sigma_free_sq;
  42 :         double prior_sd_mu_free;
  43 :         int dim_theta_i2toQ;
  44 :         int sum_C;
  45 :         std::vector<int> start_pos_gamma;
  46 :         std::vector<int> start_pos_GMMpara;
  47 : public:
  48 :     model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ(stan::io::var_context&
context__,
  49 :         std::ostream* pstream__ = 0)
  50 :         : prob_grad(0) {
  51 :         ctor_body(context__, 0, pstream__);
  52 :     }
  53 :
  54 :     model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ(stan::io::var_context&
context__,
  55 :         unsigned int random_seed__,
  56 :         std::ostream* pstream__ = 0)
  57 :         : prob_grad(0) {
  58 :         ctor_body(context__, random_seed__, pstream__);
  59 :     }
  60 :
  61 :     void ctor_body(stan::io::var_context& context__,
  62 :                    unsigned int random_seed__,
  63 :                    std::ostream* pstream__) {
  64 :         typedef double local_scalar_t__;
  65 :
  66 :         boost::ecuyer1988 base_rng__ =
  67 :           stan::services::util::create_rng(random_seed__, 0);
  68 :         (void) base_rng__;  // suppress unused var warning
  69 :
  70 :         current_statement_begin__ = -1;
  71 :
```

```
 72 :       static const char* function__ =
"model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model5baf7680f1_BL
CA_unconstrained_Givens_fixC_noZ";
 73 :       (void) function__;  // dummy to suppress unused var warning
 74 :       size_t pos__;
 75 :       (void) pos__;  // dummy to suppress unused var warning
 76 :       std::vector<int> vals_i__;
 77 :       std::vector<double> vals_r__;
 78 :       local_scalar_t__ DUMMY_VAR__(std::numeric_limits<double>::quiet_NaN());
 79 :       (void) DUMMY_VAR__;  // suppress unused var warning
 80 :
 81 :       try {
 82 :         // initialize data block variables from context__
 83 :         current_statement_begin__ = 2;
 84 :         context__.validate_dims("data initialization", "TT", "int", context__.to_vec());
 85 :         TT = int(0);
 86 :         vals_i__ = context__.vals_i("TT");
 87 :         pos__ = 0;
 88 :         TT = vals_i__[pos__++];
 89 :         check_greater_or_equal(function__, "TT", TT, 1);
 90 :
 91 :         current_statement_begin__ = 3;
 92 :         context__.validate_dims("data initialization", "Q", "int", context__.to_vec());
 93 :         Q = int(0);
 94 :         vals_i__ = context__.vals_i("Q");
 95 :         pos__ = 0;
 96 :         Q = vals_i__[pos__++];
 97 :         check_greater_or_equal(function__, "Q", Q, 1);
 98 :         check_less_or_equal(function__, "Q", Q, TT);
 99 :
100 :          current_statement_begin__ = 4;
101 :          validate_non_negative_index("C", "Q", Q);
102 :          context__.validate_dims("data initialization", "C", "int", context__.to_vec(Q));
103 :          C = std::vector<int>(Q, int(0));
104 :          vals_i__ = context__.vals_i("C");
105 :          pos__ = 0;
106 :          size_t C_k_0_max__ = Q;
107 :          for (size_t k_0__ = 0; k_0__ < C_k_0_max__; ++k_0__) {
108 :            C[k_0__] = vals_i__[pos__++];
109 :          }
110 :          size_t C_i_0_max__ = Q;
111 :          for (size_t i_0__ = 0; i_0__ < C_i_0_max__; ++i_0__) {
112 :            check_greater_or_equal(function__, "C[i_0__]", C[i_0__], 2);
113 :          }
114 :
115 :          current_statement_begin__ = 5;
116 :          context__.validate_dims("data initialization", "V", "int", context__.to_vec());
117 :          V = int(0);
118 :          vals_i__ = context__.vals_i("V");
```

```
 119 :             pos__ = 0;
 120 :             V = vals_i__[pos__++];
 121 :             check_greater_or_equal(function__, "V", V, 1);
 122 :
 123 :             current_statement_begin__ = 6;
 124 :             validate_non_negative_index("X", "V", V);
 125 :             validate_non_negative_index("X", "TT", TT);
 126 :             context__.validate_dims("data initialization", "X", "matrix_d",
context__.to_vec(V,TT));
 127 :             X = Eigen::Matrix<double, Eigen::Dynamic, Eigen::Dynamic>(V, TT);
 128 :             vals_r__ = context__.vals_r("X");
 129 :             pos__ = 0;
 130 :             size_t X_j_2_max__ = TT;
 131 :             size_t X_j_1_max__ = V;
 132 :             for (size_t j_2__ = 0; j_2__ < X_j_2_max__; ++j_2__) {
 133 :                 for (size_t j_1__ = 0; j_1__ < X_j_1_max__; ++j_1__) {
 134 :                     X(j_1__, j_2__) = vals_r__[pos__++];
 135 :                 }
 136 :             }
 137 :
 138 :             current_statement_begin__ = 8;
 139 :             context__.validate_dims("data initialization", "delta_w", "double",
context__.to_vec());
 140 :             delta_w = double(0);
 141 :             vals_r__ = context__.vals_r("delta_w");
 142 :             pos__ = 0;
 143 :             delta_w = vals_r__[pos__++];
 144 :             check_greater_or_equal(function__, "delta_w", delta_w, 0);
 145 :
 146 :             current_statement_begin__ = 9;
 147 :             context__.validate_dims("data initialization", "prior_mean_mu_free", "double",
context__.to_vec());
 148 :             prior_mean_mu_free = double(0);
 149 :             vals_r__ = context__.vals_r("prior_mean_mu_free");
 150 :             pos__ = 0;
 151 :             prior_mean_mu_free = vals_r__[pos__++];
 152 :
 153 :             current_statement_begin__ = 10;
 154 :             context__.validate_dims("data initialization", "prior_precision_mu_free", "double",
context__.to_vec());
 155 :             prior_precision_mu_free = double(0);
 156 :             vals_r__ = context__.vals_r("prior_precision_mu_free");
 157 :             pos__ = 0;
 158 :             prior_precision_mu_free = vals_r__[pos__++];
 159 :             check_greater_or_equal(function__, "prior_precision_mu_free",
prior_precision_mu_free, 0);
 160 :
 161 :             current_statement_begin__ = 11;
```

```
 162 :            context__.validate_dims("data initialization", "prior_mean_log_sigma_free_sq",
"double", context__.to_vec());
 163 :            prior_mean_log_sigma_free_sq = double(0);
 164 :            vals_r__ = context__.vals_r("prior_mean_log_sigma_free_sq");
 165 :            pos__ = 0;
 166 :            prior_mean_log_sigma_free_sq = vals_r__[pos__++];
 167 :
 168 :            current_statement_begin__ = 12;
 169 :            context__.validate_dims("data initialization", "prior_sd_log_sigma_free_sq",
"double", context__.to_vec());
 170 :            prior_sd_log_sigma_free_sq = double(0);
 171 :            vals_r__ = context__.vals_r("prior_sd_log_sigma_free_sq");
 172 :            pos__ = 0;
 173 :            prior_sd_log_sigma_free_sq = vals_r__[pos__++];
 174 :            check_greater_or_equal(function__, "prior_sd_log_sigma_free_sq",
prior_sd_log_sigma_free_sq, 0);
 175 :
 176 :
 177 :            // initialize transformed data variables
 178 :            current_statement_begin__ = 16;
 179 :            prior_sd_mu_free = double(0);
 180 :            stan::math::fill(prior_sd_mu_free, DUMMY_VAR__);
 181 :            stan::math::assign(prior_sd_mu_free,(1 /
stan::math::sqrt(prior_precision_mu_free)));
 182 :
 183 :            current_statement_begin__ = 17;
 184 :            dim_theta_i2toQ = int(0);
 185 :            stan::math::fill(dim_theta_i2toQ, std::numeric_limits<int>::min());
 186 :            stan::math::assign(dim_theta_i2toQ,((TT * (Q - 1)) - divide(((Q - 1) * (Q + 2)), 2)));
 187 :
 188 :            current_statement_begin__ = 18;
 189 :            sum_C = int(0);
 190 :            stan::math::fill(sum_C, std::numeric_limits<int>::min());
 191 :
 192 :            current_statement_begin__ = 19;
 193 :            validate_non_negative_index("start_pos_gamma", "(Q + 1)", (Q + 1));
 194 :            start_pos_gamma = std::vector<int>((Q + 1), int(0));
 195 :            stan::math::fill(start_pos_gamma, std::numeric_limits<int>::min());
 196 :
 197 :            current_statement_begin__ = 20;
 198 :            validate_non_negative_index("start_pos_GMMpara", "(Q + 1)", (Q + 1));
 199 :            start_pos_GMMpara = std::vector<int>((Q + 1), int(0));
 200 :            stan::math::fill(start_pos_GMMpara, std::numeric_limits<int>::min());
 201 :
 202 :            // execute transformed data statements
 203 :            current_statement_begin__ = 21;
 204 :            stan::model::assign(start_pos_gamma,
 205 :                        stan::model::cons_list(stan::model::index_uni(1),
stan::model::nil_index_list()),
```

```
 206 :                     1,
 207 :                     "assigning variable start_pos_gamma");
 208 :           current_statement_begin__ = 22;
 209 :           stan::model::assign(start_pos_GMMpara,
 210 :                     stan::model::cons_list(stan::model::index_uni(1),
stan::model::nil_index_list()),
 211 :                     1,
 212 :                     "assigning variable start_pos_GMMpara");
 213 :           current_statement_begin__ = 23;
 214 :           if (as_bool(logical_gt(Q, 1))) {
 215 :
 216 :             current_statement_begin__ = 24;
 217 :             for (int k = 1; k <= Q; ++k) {
 218 :
 219 :               current_statement_begin__ = 25;
 220 :               stan::model::assign(start_pos_gamma,
 221 :                         stan::model::cons_list(stan::model::index_uni((k + 1)),
stan::model::nil_index_list()),
 222 :                         ((get_base1(start_pos_gamma, k, "start_pos_gamma", 1) +
get_base1(C, k, "C", 1)) - 1),
 223 :                         "assigning variable start_pos_gamma");
 224 :               current_statement_begin__ = 26;
 225 :               stan::model::assign(start_pos_GMMpara,
 226 :                         stan::model::cons_list(stan::model::index_uni((k + 1)),
stan::model::nil_index_list()),
 227 :                         (get_base1(start_pos_GMMpara, k, "start_pos_GMMpara", 1) +
get_base1(C, k, "C", 1)),
 228 :                         "assigning variable start_pos_GMMpara");
 229 :             }
 230 :           }
 231 :           current_statement_begin__ = 29;
 232 :           stan::math::assign(sum_C, (get_base1(start_pos_GMMpara, (Q + 1),
"start_pos_GMMpara", 1) - 1));
 233 :
 234 :           // validate transformed data
 235 :
 236 :           // validate, set parameter ranges
 237 :           num_params_r__ = 0U;
 238 :           param_ranges_i__.clear();
 239 :           current_statement_begin__ = 32;
 240 :           validate_non_negative_index("theta_i1", "(TT - 1)", (TT - 1));
 241 :           num_params_r__ += (TT - 1);
 242 :           current_statement_begin__ = 33;
 243 :           validate_non_negative_index("theta_i2toQ", "dim_theta_i2toQ", dim_theta_i2toQ);
 244 :           num_params_r__ += dim_theta_i2toQ;
 245 :           current_statement_begin__ = 34;
 246 :           validate_non_negative_index("gamma_w", "(sum_C - Q)", (sum_C - Q));
 247 :           num_params_r__ += (sum_C - Q);
 248 :           current_statement_begin__ = 35;
```

```
 249 :           validate_non_negative_index("mu_free", "sum_C", sum_C);
 250 :           num_params_r__ += sum_C;
 251 :           current_statement_begin__ = 36;
 252 :           validate_non_negative_index("log_sigma_free_sq", "sum_C", sum_C);
 253 :           num_params_r__ += sum_C;
 254 :       } catch (const std::exception& e) {
 255 :           stan::lang::rethrow_located(e, current_statement_begin__, prog_reader__());
 256 :           // Next line prevents compiler griping about no return
 257 :           throw std::runtime_error("*** IF YOU SEE THIS, PLEASE REPORT A BUG ***");
 258 :       }
 259 :   }
 260 :
 261 :   ~model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ() { }
 262 :
 263 :
 264 :   void transform_inits(const stan::io::var_context& context__,
 265 :                       std::vector<int>& params_i__,
 266 :                       std::vector<double>& params_r__,
 267 :                       std::ostream* pstream__) const {
 268 :       typedef double local_scalar_t__;
 269 :       stan::io::writer<double> writer__(params_r__, params_i__);
 270 :       size_t pos__;
 271 :       (void) pos__; // dummy call to supress warning
 272 :       std::vector<double> vals_r__;
 273 :       std::vector<int> vals_i__;
 274 :
 275 :       current_statement_begin__ = 32;
 276 :       if (!(context__.contains_r("theta_i1")))
 277 :           stan::lang::rethrow_located(std::runtime_error(std::string("Variable theta_i1
missing")), current_statement_begin__, prog_reader__());
 278 :       vals_r__ = context__.vals_r("theta_i1");
 279 :       pos__ = 0U;
 280 :       validate_non_negative_index("theta_i1", "(TT - 1)", (TT - 1));
 281 :       context__.validate_dims("parameter initialization", "theta_i1", "vector_d",
context__.to_vec((TT - 1)));
 282 :       Eigen::Matrix<double, Eigen::Dynamic, 1> theta_i1((TT - 1));
 283 :       size_t theta_i1_j_1_max__ = (TT - 1);
 284 :       for (size_t j_1__ = 0; j_1__ < theta_i1_j_1_max__; ++j_1__) {
 285 :           theta_i1(j_1__) = vals_r__[pos__++];
 286 :       }
 287 :       try {
 288 :           writer__.vector_lub_unconstrain(-(stan::math::pi()), stan::math::pi(), theta_i1);
 289 :       } catch (const std::exception& e) {
 290 :           stan::lang::rethrow_located(std::runtime_error(std::string("Error transforming
variable theta_i1: ") + e.what()), current_statement_begin__, prog_reader__());
 291 :       }
 292 :
 293 :       current_statement_begin__ = 33;
 294 :       if (!(context__.contains_r("theta_i2toQ")))
```

```
 295 :             stan::lang::rethrow_located(std::runtime_error(std::string("Variable theta_i2toQ
missing")), current_statement_begin__, prog_reader__());
 296 :         vals_r__ = context__.vals_r("theta_i2toQ");
 297 :         pos__ = 0U;
 298 :         validate_non_negative_index("theta_i2toQ", "dim_theta_i2toQ", dim_theta_i2toQ);
 299 :         context__.validate_dims("parameter initialization", "theta_i2toQ", "vector_d",
context__.to_vec(dim_theta_i2toQ));
 300 :         Eigen::Matrix<double, Eigen::Dynamic, 1> theta_i2toQ(dim_theta_i2toQ);
 301 :         size_t theta_i2toQ_j_1_max__ = dim_theta_i2toQ;
 302 :         for (size_t j_1__ = 0; j_1__ < theta_i2toQ_j_1_max__; ++j_1__) {
 303 :             theta_i2toQ(j_1__) = vals_r__[pos__++];
 304 :         }
 305 :         try {
 306 :             writer__.vector_lub_unconstrain(0, stan::math::pi(), theta_i2toQ);
 307 :         } catch (const std::exception& e) {
 308 :             stan::lang::rethrow_located(std::runtime_error(std::string("Error transforming
variable theta_i2toQ: ") + e.what()), current_statement_begin__, prog_reader__());
 309 :         }
 310 :
 311 :         current_statement_begin__ = 34;
 312 :         if (!(context__.contains_r("gamma_w")))
 313 :             stan::lang::rethrow_located(std::runtime_error(std::string("Variable gamma_w
missing")), current_statement_begin__, prog_reader__());
 314 :         vals_r__ = context__.vals_r("gamma_w");
 315 :         pos__ = 0U;
 316 :         validate_non_negative_index("gamma_w", "(sum_C - Q)", (sum_C - Q));
 317 :         context__.validate_dims("parameter initialization", "gamma_w", "vector_d",
context__.to_vec((sum_C - Q)));
 318 :         Eigen::Matrix<double, Eigen::Dynamic, 1> gamma_w((sum_C - Q));
 319 :         size_t gamma_w_j_1_max__ = (sum_C - Q);
 320 :         for (size_t j_1__ = 0; j_1__ < gamma_w_j_1_max__; ++j_1__) {
 321 :             gamma_w(j_1__) = vals_r__[pos__++];
 322 :         }
 323 :         try {
 324 :             writer__.vector_lb_unconstrain(0, gamma_w);
 325 :         } catch (const std::exception& e) {
 326 :             stan::lang::rethrow_located(std::runtime_error(std::string("Error transforming
variable gamma_w: ") + e.what()), current_statement_begin__, prog_reader__());
 327 :         }
 328 :
 329 :         current_statement_begin__ = 35;
 330 :         if (!(context__.contains_r("mu_free")))
 331 :             stan::lang::rethrow_located(std::runtime_error(std::string("Variable mu_free
missing")), current_statement_begin__, prog_reader__());
 332 :         vals_r__ = context__.vals_r("mu_free");
 333 :         pos__ = 0U;
 334 :         validate_non_negative_index("mu_free", "sum_C", sum_C);
 335 :         context__.validate_dims("parameter initialization", "mu_free", "vector_d",
context__.to_vec(sum_C));
```

```
 336 :         Eigen::Matrix<double, Eigen::Dynamic, 1> mu_free(sum_C);
 337 :         size_t mu_free_j_1_max__ = sum_C;
 338 :         for (size_t j_1__ = 0; j_1__ < mu_free_j_1_max__; ++j_1__) {
 339 :           mu_free(j_1__) = vals_r__[pos__++];
 340 :         }
 341 :         try {
 342 :           writer__.vector_unconstrain(mu_free);
 343 :         } catch (const std::exception& e) {
 344 :           stan::lang::rethrow_located(std::runtime_error(std::string("Error transforming
variable mu_free: ") + e.what()), current_statement_begin__, prog_reader__());
 345 :         }
 346 :
 347 :         current_statement_begin__ = 36;
 348 :         if (!(context__.contains_r("log_sigma_free_sq")))
 349 :           stan::lang::rethrow_located(std::runtime_error(std::string("Variable
log_sigma_free_sq missing")), current_statement_begin__, prog_reader__());
 350 :         vals_r__ = context__.vals_r("log_sigma_free_sq");
 351 :         pos__ = 0U;
 352 :         validate_non_negative_index("log_sigma_free_sq", "sum_C", sum_C);
 353 :         context__.validate_dims("parameter initialization", "log_sigma_free_sq", "vector_d",
context__.to_vec(sum_C));
 354 :         Eigen::Matrix<double, Eigen::Dynamic, 1> log_sigma_free_sq(sum_C);
 355 :         size_t log_sigma_free_sq_j_1_max__ = sum_C;
 356 :         for (size_t j_1__ = 0; j_1__ < log_sigma_free_sq_j_1_max__; ++j_1__) {
 357 :           log_sigma_free_sq(j_1__) = vals_r__[pos__++];
 358 :         }
 359 :         try {
 360 :           writer__.vector_unconstrain(log_sigma_free_sq);
 361 :         } catch (const std::exception& e) {
 362 :           stan::lang::rethrow_located(std::runtime_error(std::string("Error transforming
variable log_sigma_free_sq: ") + e.what()), current_statement_begin__, prog_reader__());
 363 :         }
 364 :
 365 :         params_r__ = writer__.data_r();
 366 :         params_i__ = writer__.data_i();
 367 :     }
 368 :
 369 :     void transform_inits(const stan::io::var_context& context,
 370 :                          Eigen::Matrix<double, Eigen::Dynamic, 1>& params_r,
 371 :                          std::ostream* pstream__) const {
 372 :       std::vector<double> params_r_vec;
 373 :       std::vector<int> params_i_vec;
 374 :       transform_inits(context, params_i_vec, params_r_vec, pstream__);
 375 :       params_r.resize(params_r_vec.size());
 376 :       for (int i = 0; i < params_r.size(); ++i)
 377 :         params_r(i) = params_r_vec[i];
 378 :     }
 379 :
 380 :
```

```
 381 :    template <bool propto__, bool jacobian__, typename T__>
 382 :    T__ log_prob(std::vector<T__>& params_r__,
 383 :                 std::vector<int>& params_i__,
 384 :                 std::ostream* pstream__ = 0) const {
 385 :
 386 :      typedef T__ local_scalar_t__;
 387 :
 388 :      local_scalar_t__ DUMMY_VAR__(std::numeric_limits<double>::quiet_NaN());
 389 :      (void) DUMMY_VAR__;  // dummy to suppress unused var warning
 390 :
 391 :      T__ lp__(0.0);
 392 :      stan::math::accumulator<T__> lp_accum__;
 393 :      try {
 394 :        stan::io::reader<local_scalar_t__> in__(params_r__, params_i__);
 395 :
 396 :        // model parameters
 397 :        current_statement_begin__ = 32;
 398 :        Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> theta_i1;
 399 :        (void) theta_i1;  // dummy to suppress unused var warning
 400 :        if (jacobian__)
 401 :          theta_i1 = in__.vector_lub_constrain(-(stan::math::pi()), stan::math::pi(), (TT -
1), lp__);
 402 :        else
 403 :          theta_i1 = in__.vector_lub_constrain(-(stan::math::pi()), stan::math::pi(), (TT -
1));
 404 :
 405 :        current_statement_begin__ = 33;
 406 :        Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> theta_i2toQ;
 407 :        (void) theta_i2toQ;  // dummy to suppress unused var warning
 408 :        if (jacobian__)
 409 :          theta_i2toQ = in__.vector_lub_constrain(0, stan::math::pi(), dim_theta_i2toQ,
lp__);
 410 :        else
 411 :          theta_i2toQ = in__.vector_lub_constrain(0, stan::math::pi(), dim_theta_i2toQ);
 412 :
 413 :        current_statement_begin__ = 34;
 414 :        Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> gamma_w;
 415 :        (void) gamma_w;  // dummy to suppress unused var warning
 416 :        if (jacobian__)
 417 :          gamma_w = in__.vector_lb_constrain(0, (sum_C - Q), lp__);
 418 :        else
 419 :          gamma_w = in__.vector_lb_constrain(0, (sum_C - Q));
 420 :
 421 :        current_statement_begin__ = 35;
 422 :        Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> mu_free;
 423 :        (void) mu_free;  // dummy to suppress unused var warning
 424 :        if (jacobian__)
 425 :          mu_free = in__.vector_constrain(sum_C, lp__);
 426 :        else
```

```
 427 :             mu_free = in__.vector_constrain(sum_C);
 428 :
 429 :         current_statement_begin__ = 36;
 430 :         Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> log_sigma_free_sq;
 431 :         (void) log_sigma_free_sq;  // dummy to suppress unused var warning
 432 :         if (jacobian__)
 433 :             log_sigma_free_sq = in__.vector_constrain(sum_C, lp__);
 434 :         else
 435 :             log_sigma_free_sq = in__.vector_constrain(sum_C);
 436 :
 437 :         // transformed parameters
 438 :         current_statement_begin__ = 39;
 439 :         validate_non_negative_index("U_transpose", "TT", TT);
 440 :         validate_non_negative_index("U_transpose", "TT", TT);
 441 :         Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, Eigen::Dynamic>
U_transpose(TT, TT);
 442 :         stan::math::initialize(U_transpose, DUMMY_VAR__);
 443 :         stan::math::fill(U_transpose, DUMMY_VAR__);
 444 :         stan::math::assign(U_transpose,diag_matrix(rep_vector(1, TT)));
 445 :
 446 :         current_statement_begin__ = 40;
 447 :         validate_non_negative_index("w", "sum_C", sum_C);
 448 :         Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> w(sum_C);
 449 :         stan::math::initialize(w, DUMMY_VAR__);
 450 :         stan::math::fill(w, DUMMY_VAR__);
 451 :
 452 :         current_statement_begin__ = 44;
 453 :         validate_non_negative_index("sum_gamma", "Q", Q);
 454 :         Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> sum_gamma(Q);
 455 :         stan::math::initialize(sum_gamma, DUMMY_VAR__);
 456 :         stan::math::fill(sum_gamma, DUMMY_VAR__);
 457 :
 458 :         // transformed parameters block statements
 459 :         {
 460 :         current_statement_begin__ = 49;
 461 :         int index(0);
 462 :         (void) index;  // dummy to suppress unused var warning
 463 :         stan::math::fill(index, std::numeric_limits<int>::min());
 464 :         stan::math::assign(index,1);
 465 :
 466 :         current_statement_begin__ = 50;
 467 :         local_scalar_t__ cos1(DUMMY_VAR__);
 468 :         (void) cos1;  // dummy to suppress unused var warning
 469 :         stan::math::initialize(cos1, DUMMY_VAR__);
 470 :         stan::math::fill(cos1, DUMMY_VAR__);
 471 :
 472 :         current_statement_begin__ = 51;
 473 :         local_scalar_t__ sin1(DUMMY_VAR__);
 474 :         (void) sin1;  // dummy to suppress unused var warning
```

```
 475 :           stan::math::initialize(sin1, DUMMY_VAR__);
 476 :           stan::math::fill(sin1, DUMMY_VAR__);
 477 :
 478 :           current_statement_begin__ = 52;
 479 :           validate_non_negative_index("Ui_tmp", "TT", TT);
 480 :           Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> Ui_tmp(TT);
 481 :           stan::math::initialize(Ui_tmp, DUMMY_VAR__);
 482 :           stan::math::fill(Ui_tmp, DUMMY_VAR__);
 483 :
 484 :           current_statement_begin__ = 53;
 485 :           validate_non_negative_index("Uj_tmp", "TT", TT);
 486 :           Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> Uj_tmp(TT);
 487 :           stan::math::initialize(Uj_tmp, DUMMY_VAR__);
 488 :           stan::math::fill(Uj_tmp, DUMMY_VAR__);
 489 :
 490 :
 491 :           current_statement_begin__ = 55;
 492 :           for (int k = 1; k <= Q; ++k) {
 493 :             {
 494 :               current_statement_begin__ = 56;
 495 :               validate_non_negative_index("gamma_w_seg", "(get_base1(C, k, \"C\", 1) - 1)",
(get_base1(C, k, "C", 1) - 1));
 496 :               Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1>
gamma_w_seg((get_base1(C, k, "C", 1) - 1));
 497 :               stan::math::initialize(gamma_w_seg, DUMMY_VAR__);
 498 :               stan::math::fill(gamma_w_seg, DUMMY_VAR__);
 499 :               stan::math::assign(gamma_w_seg,segment(gamma_w,
get_base1(start_pos_gamma, k, "start_pos_gamma", 1), (get_base1(C, k, "C", 1) - 1)));
 500 :
 501 :
 502 :               current_statement_begin__ = 60;
 503 :               if (as_bool(logical_eq(k, 1))) {
 504 :
 505 :                 current_statement_begin__ = 61;
 506 :                 for (int j = (k + 1); j <= TT; ++j) {
 507 :
 508 :                   current_statement_begin__ = 62;
 509 :                   stan::math::assign(cos1, stan::math::cos(get_base1(theta_i1, index,
"theta_i1", 1)));
 510 :                   current_statement_begin__ = 63;
 511 :                   stan::math::assign(sin1, stan::math::sin(get_base1(theta_i1, index,
"theta_i1", 1)));
 512 :                   current_statement_begin__ = 64;
 513 :                   stan::math::assign(Ui_tmp, subtract(multiply(cos1,
stan::model::rvalue(U_transpose, stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(k), stan::model::nil_index_list())), "U_transpose")),
multiply(sin1, stan::model::rvalue(U_transpose,
stan::model::cons_list(stan::model::index_omni(),
```

```
stan::model::cons_list(stan::model::index_uni(j), stan::model::nil_index_list())),
"U_transpose"))));
 514 :                     current_statement_begin__ = 65;
 515 :                     stan::math::assign(Uj_tmp, add(multiply(sin1,
stan::model::rvalue(U_transpose, stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(k), stan::model::nil_index_list())), "U_transpose")),
multiply(cos1, stan::model::rvalue(U_transpose,
stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(j), stan::model::nil_index_list())),
"U_transpose"))));
 516 :                     current_statement_begin__ = 66;
 517 :                     stan::model::assign(U_transpose,
 518 :                             stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(k), stan::model::nil_index_list())),
 519 :                             Ui_tmp,
 520 :                             "assigning variable U_transpose");
 521 :                     current_statement_begin__ = 67;
 522 :                     stan::model::assign(U_transpose,
 523 :                             stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(j), stan::model::nil_index_list())),
 524 :                             Uj_tmp,
 525 :                             "assigning variable U_transpose");
 526 :                     current_statement_begin__ = 68;
 527 :                     stan::math::assign(index, (index + 1));
 528 :                 }
 529 :                 current_statement_begin__ = 70;
 530 :                 stan::math::assign(index, 1);
 531 :             } else {
 532 :
 533 :                 current_statement_begin__ = 72;
 534 :                 for (int j = (k + 1); j <= TT; ++j) {
 535 :
 536 :                     current_statement_begin__ = 73;
 537 :                     stan::math::assign(cos1, stan::math::cos(get_base1(theta_i2toQ, index,
"theta_i2toQ", 1)));
 538 :                     current_statement_begin__ = 74;
 539 :                     stan::math::assign(sin1, stan::math::sin(get_base1(theta_i2toQ, index,
"theta_i2toQ", 1)));
 540 :                     current_statement_begin__ = 75;
 541 :                     stan::math::assign(Ui_tmp, subtract(multiply(cos1,
stan::model::rvalue(U_transpose, stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(k), stan::model::nil_index_list())), "U_transpose")),
multiply(sin1, stan::model::rvalue(U_transpose,
stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(j), stan::model::nil_index_list())),
"U_transpose"))));
 542 :                     current_statement_begin__ = 76;
 543 :                     stan::math::assign(Uj_tmp, add(multiply(sin1,
stan::model::rvalue(U_transpose, stan::model::cons_list(stan::model::index_omni(),
```

```
stan::model::cons_list(stan::model::index_uni(k), stan::model::nil_index_list())), "U_transpose")),
multiply(cos1, stan::model::rvalue(U_transpose,
stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(j), stan::model::nil_index_list())),
"U_transpose"))));
 544 :                     current_statement_begin__ = 77;
 545 :                     stan::model::assign(U_transpose,
 546 :                             stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(k), stan::model::nil_index_list())),
 547 :                             Ui_tmp,
 548 :                             "assigning variable U_transpose");
 549 :                     current_statement_begin__ = 78;
 550 :                     stan::model::assign(U_transpose,
 551 :                             stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(j), stan::model::nil_index_list())),
 552 :                             Uj_tmp,
 553 :                             "assigning variable U_transpose");
 554 :                     current_statement_begin__ = 79;
 555 :                     stan::math::assign(index, (index + 1));
 556 :                   }
 557 :                 }
 558 :                 current_statement_begin__ = 83;
 559 :                 stan::model::assign(sum_gamma,
 560 :                         stan::model::cons_list(stan::model::index_uni(k),
stan::model::nil_index_list()),
 561 :                         (1 + sum(gamma_w_seg)),
 562 :                         "assigning variable sum_gamma");
 563 :                 current_statement_begin__ = 84;
 564 :                 stan::model::assign(w,
 565 :
stan::model::cons_list(stan::model::index_min_max(get_base1(start_pos_GMMpara, k,
"start_pos_GMMpara", 1), (get_base1(start_pos_GMMpara, (k + 1), "start_pos_GMMpara", 1) -
1)), stan::model::nil_index_list()),
 566 :                         elt_divide(append_row(1.0, gamma_w_seg), get_base1(sum_gamma,
k, "sum_gamma", 1)),
 567 :                         "assigning variable w");
 568 :                 }
 569 :             }
 570 :             }
 571 :
 572 :             // validate transformed parameters
 573 :             const char* function__ = "validate transformed params";
 574 :             (void) function__;  // dummy to suppress unused var warning
 575 :
 576 :             current_statement_begin__ = 39;
 577 :             size_t U_transpose_j_1_max__ = TT;
 578 :             size_t U_transpose_j_2_max__ = TT;
 579 :             for (size_t j_1__ = 0; j_1__ < U_transpose_j_1_max__; ++j_1__) {
 580 :                 for (size_t j_2__ = 0; j_2__ < U_transpose_j_2_max__; ++j_2__) {
```

```
 581 :                     if (stan::math::is_uninitialized(U_transpose(j_1__, j_2__))) {
 582 :                       std::stringstream msg__;
 583 :                       msg__ << "Undefined transformed parameter: U_transpose" << "(" <<
j_1__ << ", " << j_2__ << ")";
 584 :                       stan::lang::rethrow_located(std::runtime_error(std::string("Error initializing
variable U_transpose: ") + msg__.str()), current_statement_begin__, prog_reader__());
 585 :                     }
 586 :                   }
 587 :               }
 588 :               current_statement_begin__ = 40;
 589 :               size_t w_j_1_max__ = sum_C;
 590 :               for (size_t j_1__ = 0; j_1__ < w_j_1_max__; ++j_1__) {
 591 :                   if (stan::math::is_uninitialized(w(j_1__))) {
 592 :                       std::stringstream msg__;
 593 :                       msg__ << "Undefined transformed parameter: w" << "(" << j_1__ << ")";
 594 :                       stan::lang::rethrow_located(std::runtime_error(std::string("Error initializing
variable w: ") + msg__.str()), current_statement_begin__, prog_reader__());
 595 :                   }
 596 :               }
 597 :               check_greater_or_equal(function__, "w", w, 0);
 598 :
 599 :               current_statement_begin__ = 44;
 600 :               size_t sum_gamma_j_1_max__ = Q;
 601 :               for (size_t j_1__ = 0; j_1__ < sum_gamma_j_1_max__; ++j_1__) {
 602 :                   if (stan::math::is_uninitialized(sum_gamma(j_1__))) {
 603 :                       std::stringstream msg__;
 604 :                       msg__ << "Undefined transformed parameter: sum_gamma" << "(" << j_1__
<< ")";
 605 :                       stan::lang::rethrow_located(std::runtime_error(std::string("Error initializing
variable sum_gamma: ") + msg__.str()), current_statement_begin__, prog_reader__());
 606 :                   }
 607 :               }
 608 :
 609 :               // model body
 610 :               {
 611 :               current_statement_begin__ = 89;
 612 :               validate_non_negative_index("S", "V", V);
 613 :               validate_non_negative_index("S", "Q", Q);
 614 :               Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, Eigen::Dynamic> S(V, Q);
 615 :               stan::math::initialize(S, DUMMY_VAR__);
 616 :               stan::math::fill(S, DUMMY_VAR__);
 617 :               stan::math::assign(S,multiply(X, stan::model::rvalue(U_transpose,
stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_min_max(1, Q), stan::model::nil_index_list())),
"U_transpose")));
 618 :
 619 :
 620 :               current_statement_begin__ = 95;
 621 :               for (int k = 1; k <= Q; ++k) {
```

```
 622 :             {
 623 :             current_statement_begin__ = 96;
 624 :             validate_non_negative_index("w_seg", "get_base1(C, k, \"C\", 1)",
get_base1(C, k, "C", 1));
 625 :             Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> w_seg(get_base1(C, k,
"C", 1));
 626 :             stan::math::initialize(w_seg, DUMMY_VAR__);
 627 :             stan::math::fill(w_seg, DUMMY_VAR__);
 628 :             stan::math::assign(w_seg,segment(w, get_base1(start_pos_GMMpara, k,
"start_pos_GMMpara", 1), get_base1(C, k, "C", 1)));
 629 :
 630 :             current_statement_begin__ = 97;
 631 :             validate_non_negative_index("ln_w_seg", "get_base1(C, k, \"C\", 1)",
get_base1(C, k, "C", 1));
 632 :             Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> ln_w_seg(get_base1(C, k,
"C", 1));
 633 :             stan::math::initialize(ln_w_seg, DUMMY_VAR__);
 634 :             stan::math::fill(ln_w_seg, DUMMY_VAR__);
 635 :             stan::math::assign(ln_w_seg,stan::math::log(w_seg));
 636 :
 637 :             current_statement_begin__ = 98;
 638 :             validate_non_negative_index("mu_free_seg", "get_base1(C, k, \"C\", 1)",
get_base1(C, k, "C", 1));
 639 :             Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1>
mu_free_seg(get_base1(C, k, "C", 1));
 640 :             stan::math::initialize(mu_free_seg, DUMMY_VAR__);
 641 :             stan::math::fill(mu_free_seg, DUMMY_VAR__);
 642 :             stan::math::assign(mu_free_seg,segment(mu_free,
get_base1(start_pos_GMMpara, k, "start_pos_GMMpara", 1), get_base1(C, k, "C", 1)));
 643 :
 644 :             current_statement_begin__ = 99;
 645 :             validate_non_negative_index("log_sigma_free_sq_seg", "get_base1(C, k, \"C\",
1)", get_base1(C, k, "C", 1));
 646 :             Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1>
log_sigma_free_sq_seg(get_base1(C, k, "C", 1));
 647 :             stan::math::initialize(log_sigma_free_sq_seg, DUMMY_VAR__);
 648 :             stan::math::fill(log_sigma_free_sq_seg, DUMMY_VAR__);
 649 :             stan::math::assign(log_sigma_free_sq_seg,segment(log_sigma_free_sq,
get_base1(start_pos_GMMpara, k, "start_pos_GMMpara", 1), get_base1(C, k, "C", 1)));
 650 :
 651 :             current_statement_begin__ = 100;
 652 :             validate_non_negative_index("sigma_free_seg", "get_base1(C, k, \"C\", 1)",
get_base1(C, k, "C", 1));
 653 :             Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1>
sigma_free_seg(get_base1(C, k, "C", 1));
 654 :             stan::math::initialize(sigma_free_seg, DUMMY_VAR__);
 655 :             stan::math::fill(sigma_free_seg, DUMMY_VAR__);
 656 :
stan::math::assign(sigma_free_seg,stan::math::exp(divide(log_sigma_free_sq_seg, 2)));
```

```
 657 :
 658 :             current_statement_begin__ = 102;
 659 :             validate_non_negative_index("loglik_v", "V", V);
 660 :             Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> loglik_v(V);
 661 :             stan::math::initialize(loglik_v, DUMMY_VAR__);
 662 :             stan::math::fill(loglik_v, DUMMY_VAR__);
 663 :
 664 :
 665 :             current_statement_begin__ = 108;
 666 :             lp_accum__.add(dirichlet_log<propto__>(w_seg, rep_vector(delta_w,
get_base1(C, k, "C", 1))));
 667 :             current_statement_begin__ = 113;
 668 :             lp_accum__.add((-(get_base1(C, k, "C", 1)) *
stan::math::log(get_base1(sum_gamma, k, "sum_gamma", 1))));
 669 :             current_statement_begin__ = 120;
 670 :             lp_accum__.add(normal_log<propto__>(mu_free_seg, prior_mean_mu_free,
prior_sd_mu_free));
 671 :             current_statement_begin__ = 127;
 672 :             lp_accum__.add(normal_log<propto__>(log_sigma_free_sq_seg,
prior_mean_log_sigma_free_sq, prior_sd_log_sigma_free_sq));
 673 :             current_statement_begin__ = 134;
 674 :             for (int v = 1; v <= V; ++v) {
 675 :               {
 676 :               current_statement_begin__ = 135;
 677 :               validate_non_negative_index("lps", "get_base1(C, k, \"C\", 1)", get_base1(C,
k, "C", 1));
 678 :               Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> lps(get_base1(C, k, "C",
1));
 679 :               stan::math::initialize(lps, DUMMY_VAR__);
 680 :               stan::math::fill(lps, DUMMY_VAR__);
 681 :               stan::math::assign(lps,ln_w_seg);
 682 :
 683 :
 684 :               current_statement_begin__ = 136;
 685 :               for (int cc = 1; cc <= get_base1(C, k, "C", 1); ++cc) {
 686 :
 687 :                 current_statement_begin__ = 137;
 688 :                 stan::model::assign(lps,
 689 :                           stan::model::cons_list(stan::model::index_uni(cc),
stan::model::nil_index_list()),
 690 :                           (stan::model::rvalue(lps,
stan::model::cons_list(stan::model::index_uni(cc), stan::model::nil_index_list()), "lps") +
normal_log(get_base1(S, v, k, "S", 1), get_base1(mu_free_seg, cc, "mu_free_seg", 1),
get_base1(sigma_free_seg, cc, "sigma_free_seg", 1))),
 691 :                           "assigning variable lps");
 692 :               }
 693 :               current_statement_begin__ = 139;
 694 :               stan::model::assign(loglik_v,
```

```
 695 :                             stan::model::cons_list(stan::model::index_uni(v),
stan::model::nil_index_list()),
 696 :                             log_sum_exp(lps),
 697 :                             "assigning variable loglik_v");
 698 :                     }
 699 :                 }
 700 :                 current_statement_begin__ = 151;
 701 :                 lp_accum__.add(sum(loglik_v));
 702 :                 }
 703 :             }
 704 :             }
 705 :
 706 :         } catch (const std::exception& e) {
 707 :             stan::lang::rethrow_located(e, current_statement_begin__, prog_reader__());
 708 :             // Next line prevents compiler griping about no return
 709 :             throw std::runtime_error("*** IF YOU SEE THIS, PLEASE REPORT A BUG ***");
 710 :         }
 711 :
 712 :         lp_accum__.add(lp__);
 713 :         return lp_accum__.sum();
 714 :
 715 :     } // log_prob()
 716 :
 717 :     template <bool propto, bool jacobian, typename T_>
 718 :     T_ log_prob(Eigen::Matrix<T_,Eigen::Dynamic,1>& params_r,
 719 :                 std::ostream* pstream = 0) const {
 720 :       std::vector<T_> vec_params_r;
 721 :       vec_params_r.reserve(params_r.size());
 722 :       for (int i = 0; i < params_r.size(); ++i)
 723 :         vec_params_r.push_back(params_r(i));
 724 :       std::vector<int> vec_params_i;
 725 :       return log_prob<propto,jacobian,T_>(vec_params_r, vec_params_i, pstream);
 726 :     }
 727 :
 728 :
 729 :     void get_param_names(std::vector<std::string>& names__) const {
 730 :         names__.resize(0);
 731 :         names__.push_back("theta_i1");
 732 :         names__.push_back("theta_i2toQ");
 733 :         names__.push_back("gamma_w");
 734 :         names__.push_back("mu_free");
 735 :         names__.push_back("log_sigma_free_sq");
 736 :         names__.push_back("U_transpose");
 737 :         names__.push_back("w");
 738 :         names__.push_back("sum_gamma");
 739 :     }
 740 :
 741 :
 742 :     void get_dims(std::vector<std::vector<size_t> >& dimss__) const {
```

```
 743 :         dimss__.resize(0);
 744 :         std::vector<size_t> dims__;
 745 :         dims__.resize(0);
 746 :         dims__.push_back((TT - 1));
 747 :         dimss__.push_back(dims__);
 748 :         dims__.resize(0);
 749 :         dims__.push_back(dim_theta_i2toQ);
 750 :         dimss__.push_back(dims__);
 751 :         dims__.resize(0);
 752 :         dims__.push_back((sum_C - Q));
 753 :         dimss__.push_back(dims__);
 754 :         dims__.resize(0);
 755 :         dims__.push_back(sum_C);
 756 :         dimss__.push_back(dims__);
 757 :         dims__.resize(0);
 758 :         dims__.push_back(sum_C);
 759 :         dimss__.push_back(dims__);
 760 :         dims__.resize(0);
 761 :         dims__.push_back(TT);
 762 :         dims__.push_back(TT);
 763 :         dimss__.push_back(dims__);
 764 :         dims__.resize(0);
 765 :         dims__.push_back(sum_C);
 766 :         dimss__.push_back(dims__);
 767 :         dims__.resize(0);
 768 :         dims__.push_back(Q);
 769 :         dimss__.push_back(dims__);
 770 :     }
 771 :
 772 :     template <typename RNG>
 773 :     void write_array(RNG& base_rng__,
 774 :                      std::vector<double>& params_r__,
 775 :                      std::vector<int>& params_i__,
 776 :                      std::vector<double>& vars__,
 777 :                      bool include_tparams__ = true,
 778 :                      bool include_gqs__ = true,
 779 :                      std::ostream* pstream__ = 0) const {
 780 :         typedef double local_scalar_t__;
 781 :
 782 :         vars__.resize(0);
 783 :         stan::io::reader<local_scalar_t__> in__(params_r__, params_i__);
 784 :         static const char* function__ =
"model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::write_array";
 785 :         (void) function__;  // dummy to suppress unused var warning
 786 :
 787 :         // read-transform, write parameters
 788 :         Eigen::Matrix<double, Eigen::Dynamic, 1> theta_i1 = in__.vector_lub_constrain(-
(stan::math::pi()), stan::math::pi(), (TT - 1));
 789 :         size_t theta_i1_j_1_max__ = (TT - 1);
```

```
 790 :       for (size_t j_1__ = 0; j_1__ < theta_i1_j_1_max__; ++j_1__) {
 791 :         vars__.push_back(theta_i1(j_1__));
 792 :       }
 793 :
 794 :       Eigen::Matrix<double, Eigen::Dynamic, 1> theta_i2toQ =
in__.vector_lub_constrain(0, stan::math::pi(), dim_theta_i2toQ);
 795 :       size_t theta_i2toQ_j_1_max__ = dim_theta_i2toQ;
 796 :       for (size_t j_1__ = 0; j_1__ < theta_i2toQ_j_1_max__; ++j_1__) {
 797 :         vars__.push_back(theta_i2toQ(j_1__));
 798 :       }
 799 :
 800 :       Eigen::Matrix<double, Eigen::Dynamic, 1> gamma_w = in__.vector_lb_constrain(0,
(sum_C - Q));
 801 :       size_t gamma_w_j_1_max__ = (sum_C - Q);
 802 :       for (size_t j_1__ = 0; j_1__ < gamma_w_j_1_max__; ++j_1__) {
 803 :         vars__.push_back(gamma_w(j_1__));
 804 :       }
 805 :
 806 :       Eigen::Matrix<double, Eigen::Dynamic, 1> mu_free = in__.vector_constrain(sum_C);
 807 :       size_t mu_free_j_1_max__ = sum_C;
 808 :       for (size_t j_1__ = 0; j_1__ < mu_free_j_1_max__; ++j_1__) {
 809 :         vars__.push_back(mu_free(j_1__));
 810 :       }
 811 :
 812 :       Eigen::Matrix<double, Eigen::Dynamic, 1> log_sigma_free_sq =
in__.vector_constrain(sum_C);
 813 :       size_t log_sigma_free_sq_j_1_max__ = sum_C;
 814 :       for (size_t j_1__ = 0; j_1__ < log_sigma_free_sq_j_1_max__; ++j_1__) {
 815 :         vars__.push_back(log_sigma_free_sq(j_1__));
 816 :       }
 817 :
 818 :       double lp__ = 0.0;
 819 :       (void) lp__;  // dummy to suppress unused var warning
 820 :       stan::math::accumulator<double> lp_accum__;
 821 :
 822 :       local_scalar_t__ DUMMY_VAR__(std::numeric_limits<double>::quiet_NaN());
 823 :       (void) DUMMY_VAR__;  // suppress unused var warning
 824 :
 825 :       if (!include_tparams__ && !include_gqs__) return;
 826 :
 827 :       try {
 828 :         // declare and define transformed parameters
 829 :         current_statement_begin__ = 39;
 830 :         validate_non_negative_index("U_transpose", "TT", TT);
 831 :         validate_non_negative_index("U_transpose", "TT", TT);
 832 :         Eigen::Matrix<double, Eigen::Dynamic, Eigen::Dynamic> U_transpose(TT, TT);
 833 :         stan::math::initialize(U_transpose, DUMMY_VAR__);
 834 :         stan::math::fill(U_transpose, DUMMY_VAR__);
 835 :         stan::math::assign(U_transpose,diag_matrix(rep_vector(1, TT)));
```

```
836 :
837 :         current_statement_begin__ = 40;
838 :         validate_non_negative_index("w", "sum_C", sum_C);
839 :         Eigen::Matrix<double, Eigen::Dynamic, 1> w(sum_C);
840 :         stan::math::initialize(w, DUMMY_VAR__);
841 :         stan::math::fill(w, DUMMY_VAR__);
842 :
843 :         current_statement_begin__ = 44;
844 :         validate_non_negative_index("sum_gamma", "Q", Q);
845 :         Eigen::Matrix<double, Eigen::Dynamic, 1> sum_gamma(Q);
846 :         stan::math::initialize(sum_gamma, DUMMY_VAR__);
847 :         stan::math::fill(sum_gamma, DUMMY_VAR__);
848 :
849 :         // do transformed parameters statements
850 :         {
851 :         current_statement_begin__ = 49;
852 :         int index(0);
853 :         (void) index;  // dummy to suppress unused var warning
854 :         stan::math::fill(index, std::numeric_limits<int>::min());
855 :         stan::math::assign(index,1);
856 :
857 :         current_statement_begin__ = 50;
858 :         local_scalar_t__ cos1(DUMMY_VAR__);
859 :         (void) cos1;  // dummy to suppress unused var warning
860 :         stan::math::initialize(cos1, DUMMY_VAR__);
861 :         stan::math::fill(cos1, DUMMY_VAR__);
862 :
863 :         current_statement_begin__ = 51;
864 :         local_scalar_t__ sin1(DUMMY_VAR__);
865 :         (void) sin1;  // dummy to suppress unused var warning
866 :         stan::math::initialize(sin1, DUMMY_VAR__);
867 :         stan::math::fill(sin1, DUMMY_VAR__);
868 :
869 :         current_statement_begin__ = 52;
870 :         validate_non_negative_index("Ui_tmp", "TT", TT);
871 :         Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> Ui_tmp(TT);
872 :         stan::math::initialize(Ui_tmp, DUMMY_VAR__);
873 :         stan::math::fill(Ui_tmp, DUMMY_VAR__);
874 :
875 :         current_statement_begin__ = 53;
876 :         validate_non_negative_index("Uj_tmp", "TT", TT);
877 :         Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> Uj_tmp(TT);
878 :         stan::math::initialize(Uj_tmp, DUMMY_VAR__);
879 :         stan::math::fill(Uj_tmp, DUMMY_VAR__);
880 :
881 :
882 :         current_statement_begin__ = 55;
883 :         for (int k = 1; k <= Q; ++k) {
884 :           {
```

```
 885 :             current_statement_begin__ = 56;
 886 :             validate_non_negative_index("gamma_w_seg", "(get_base1(C, k, \"C\", 1) - 1)",
(get_base1(C, k, "C", 1) - 1));
 887 :             Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1>
gamma_w_seg((get_base1(C, k, "C", 1) - 1));
 888 :             stan::math::initialize(gamma_w_seg, DUMMY_VAR__);
 889 :             stan::math::fill(gamma_w_seg, DUMMY_VAR__);
 890 :             stan::math::assign(gamma_w_seg,segment(gamma_w,
get_base1(start_pos_gamma, k, "start_pos_gamma", 1), (get_base1(C, k, "C", 1) - 1)));
 891 :
 892 :
 893 :             current_statement_begin__ = 60;
 894 :             if (as_bool(logical_eq(k, 1))) {
 895 :
 896 :                 current_statement_begin__ = 61;
 897 :                 for (int j = (k + 1); j <= TT; ++j) {
 898 :
 899 :                     current_statement_begin__ = 62;
 900 :                     stan::math::assign(cos1, stan::math::cos(get_base1(theta_i1, index,
"theta_i1", 1)));
 901 :                     current_statement_begin__ = 63;
 902 :                     stan::math::assign(sin1, stan::math::sin(get_base1(theta_i1, index,
"theta_i1", 1)));
 903 :                     current_statement_begin__ = 64;
 904 :                     stan::math::assign(Ui_tmp, subtract(multiply(cos1,
stan::model::rvalue(U_transpose, stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(k), stan::model::nil_index_list())), "U_transpose")),
multiply(sin1, stan::model::rvalue(U_transpose,
stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(j), stan::model::nil_index_list())),
"U_transpose"))));
 905 :                     current_statement_begin__ = 65;
 906 :                     stan::math::assign(Uj_tmp, add(multiply(sin1,
stan::model::rvalue(U_transpose, stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(k), stan::model::nil_index_list())), "U_transpose")),
multiply(cos1, stan::model::rvalue(U_transpose,
stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(j), stan::model::nil_index_list())),
"U_transpose"))));
 907 :                     current_statement_begin__ = 66;
 908 :                     stan::model::assign(U_transpose,
 909 :                             stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(k), stan::model::nil_index_list())),
 910 :                             Ui_tmp,
 911 :                             "assigning variable U_transpose");
 912 :                     current_statement_begin__ = 67;
 913 :                     stan::model::assign(U_transpose,
 914 :                             stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(j), stan::model::nil_index_list())),
```

```
 915 :                             Uj_tmp,
 916 :                             "assigning variable U_transpose");
 917 :                     current_statement_begin__ = 68;
 918 :                     stan::math::assign(index, (index + 1));
 919 :                 }
 920 :                 current_statement_begin__ = 70;
 921 :                 stan::math::assign(index, 1);
 922 :             } else {
 923 :
 924 :                 current_statement_begin__ = 72;
 925 :                 for (int j = (k + 1); j <= TT; ++j) {
 926 :
 927 :                     current_statement_begin__ = 73;
 928 :                     stan::math::assign(cos1, stan::math::cos(get_base1(theta_i2toQ, index,
"theta_i2toQ", 1)));
 929 :                     current_statement_begin__ = 74;
 930 :                     stan::math::assign(sin1, stan::math::sin(get_base1(theta_i2toQ, index,
"theta_i2toQ", 1)));
 931 :                     current_statement_begin__ = 75;
 932 :                     stan::math::assign(Ui_tmp, subtract(multiply(cos1,
stan::model::rvalue(U_transpose, stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(k), stan::model::nil_index_list())), "U_transpose")),
multiply(sin1, stan::model::rvalue(U_transpose,
stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(j), stan::model::nil_index_list())),
"U_transpose"))));
 933 :                     current_statement_begin__ = 76;
 934 :                     stan::math::assign(Uj_tmp, add(multiply(sin1,
stan::model::rvalue(U_transpose, stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(k), stan::model::nil_index_list())), "U_transpose")),
multiply(cos1, stan::model::rvalue(U_transpose,
stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(j), stan::model::nil_index_list())),
"U_transpose"))));
 935 :                     current_statement_begin__ = 77;
 936 :                     stan::model::assign(U_transpose,
 937 :                             stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(k), stan::model::nil_index_list())),
 938 :                             Ui_tmp,
 939 :                             "assigning variable U_transpose");
 940 :                     current_statement_begin__ = 78;
 941 :                     stan::model::assign(U_transpose,
 942 :                             stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(j), stan::model::nil_index_list())),
 943 :                             Uj_tmp,
 944 :                             "assigning variable U_transpose");
 945 :                     current_statement_begin__ = 79;
 946 :                     stan::math::assign(index, (index + 1));
 947 :                 }
```

```
 948 :             }
 949 :             current_statement_begin__ = 83;
 950 :             stan::model::assign(sum_gamma,
 951 :                         stan::model::cons_list(stan::model::index_uni(k),
stan::model::nil_index_list()),
 952 :                         (1 + sum(gamma_w_seg)),
 953 :                         "assigning variable sum_gamma");
 954 :             current_statement_begin__ = 84;
 955 :             stan::model::assign(w,
 956 :
stan::model::cons_list(stan::model::index_min_max(get_base1(start_pos_GMMpara, k,
"start_pos_GMMpara", 1), (get_base1(start_pos_GMMpara, (k + 1), "start_pos_GMMpara", 1) -
1)), stan::model::nil_index_list()),
 957 :                         elt_divide(append_row(1.0, gamma_w_seg), get_base1(sum_gamma,
k, "sum_gamma", 1)),
 958 :                         "assigning variable w");
 959 :             }
 960 :         }
 961 :         }
 962 :
 963 :         if (!include_gqs__ && !include_tparams__) return;
 964 :         // validate transformed parameters
 965 :         const char* function__ = "validate transformed params";
 966 :         (void) function__;  // dummy to suppress unused var warning
 967 :
 968 :         current_statement_begin__ = 40;
 969 :         check_greater_or_equal(function__, "w", w, 0);
 970 :
 971 :         // write transformed parameters
 972 :         if (include_tparams__) {
 973 :             size_t U_transpose_j_2_max__ = TT;
 974 :             size_t U_transpose_j_1_max__ = TT;
 975 :             for (size_t j_2__ = 0; j_2__ < U_transpose_j_2_max__; ++j_2__) {
 976 :                 for (size_t j_1__ = 0; j_1__ < U_transpose_j_1_max__; ++j_1__) {
 977 :                     vars__.push_back(U_transpose(j_1__, j_2__));
 978 :                 }
 979 :             }
 980 :             size_t w_j_1_max__ = sum_C;
 981 :             for (size_t j_1__ = 0; j_1__ < w_j_1_max__; ++j_1__) {
 982 :                 vars__.push_back(w(j_1__));
 983 :             }
 984 :             size_t sum_gamma_j_1_max__ = Q;
 985 :             for (size_t j_1__ = 0; j_1__ < sum_gamma_j_1_max__; ++j_1__) {
 986 :                 vars__.push_back(sum_gamma(j_1__));
 987 :             }
 988 :         }
 989 :         if (!include_gqs__) return;
 990 :       } catch (const std::exception& e) {
 991 :         stan::lang::rethrow_located(e, current_statement_begin__, prog_reader__());
```

```
 992 :           // Next line prevents compiler griping about no return
 993 :           throw std::runtime_error("*** IF YOU SEE THIS, PLEASE REPORT A BUG ***");
 994 :         }
 995 :     }
 996 :
 997 :     template <typename RNG>
 998 :     void write_array(RNG& base_rng,
 999 :                      Eigen::Matrix<double,Eigen::Dynamic,1>& params_r,
1000 :                      Eigen::Matrix<double,Eigen::Dynamic,1>& vars,
1001 :                      bool include_tparams = true,
1002 :                      bool include_gqs = true,
1003 :                      std::ostream* pstream = 0) const {
1004 :       std::vector<double> params_r_vec(params_r.size());
1005 :       for (int i = 0; i < params_r.size(); ++i)
1006 :         params_r_vec[i] = params_r(i);
1007 :       std::vector<double> vars_vec;
1008 :       std::vector<int> params_i_vec;
1009 :       write_array(base_rng, params_r_vec, params_i_vec, vars_vec, include_tparams,
include_gqs, pstream);
1010 :       vars.resize(vars_vec.size());
1011 :       for (int i = 0; i < vars.size(); ++i)
1012 :         vars(i) = vars_vec[i];
1013 :     }
1014 :
1015 :     static std::string model_name() {
1016 :         return "model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ";
1017 :     }
1018 :
1019 :
1020 :     void constrained_param_names(std::vector<std::string>& param_names__,
1021 :                                  bool include_tparams__ = true,
1022 :                                  bool include_gqs__ = true) const {
1023 :         std::stringstream param_name_stream__;
1024 :         size_t theta_i1_j_1_max__ = (TT - 1);
1025 :         for (size_t j_1__ = 0; j_1__ < theta_i1_j_1_max__; ++j_1__) {
1026 :             param_name_stream__.str(std::string());
1027 :             param_name_stream__ << "theta_i1" << '.' << j_1__ + 1;
1028 :             param_names__.push_back(param_name_stream__.str());
1029 :         }
1030 :         size_t theta_i2toQ_j_1_max__ = dim_theta_i2toQ;
1031 :         for (size_t j_1__ = 0; j_1__ < theta_i2toQ_j_1_max__; ++j_1__) {
1032 :             param_name_stream__.str(std::string());
1033 :             param_name_stream__ << "theta_i2toQ" << '.' << j_1__ + 1;
1034 :             param_names__.push_back(param_name_stream__.str());
1035 :         }
1036 :         size_t gamma_w_j_1_max__ = (sum_C - Q);
1037 :         for (size_t j_1__ = 0; j_1__ < gamma_w_j_1_max__; ++j_1__) {
1038 :             param_name_stream__.str(std::string());
1039 :             param_name_stream__ << "gamma_w" << '.' << j_1__ + 1;
```

```
1040 :           param_names__.push_back(param_name_stream__.str());
1041 :       }
1042 :       size_t mu_free_j_1_max__ = sum_C;
1043 :       for (size_t j_1__ = 0; j_1__ < mu_free_j_1_max__; ++j_1__) {
1044 :           param_name_stream__.str(std::string());
1045 :           param_name_stream__ << "mu_free" << '.' << j_1__ + 1;
1046 :           param_names__.push_back(param_name_stream__.str());
1047 :       }
1048 :       size_t log_sigma_free_sq_j_1_max__ = sum_C;
1049 :       for (size_t j_1__ = 0; j_1__ < log_sigma_free_sq_j_1_max__; ++j_1__) {
1050 :           param_name_stream__.str(std::string());
1051 :           param_name_stream__ << "log_sigma_free_sq" << '.' << j_1__ + 1;
1052 :           param_names__.push_back(param_name_stream__.str());
1053 :       }
1054 :
1055 :       if (!include_gqs__ && !include_tparams__) return;
1056 :
1057 :       if (include_tparams__) {
1058 :           size_t U_transpose_j_2_max__ = TT;
1059 :           size_t U_transpose_j_1_max__ = TT;
1060 :           for (size_t j_2__ = 0; j_2__ < U_transpose_j_2_max__; ++j_2__) {
1061 :               for (size_t j_1__ = 0; j_1__ < U_transpose_j_1_max__; ++j_1__) {
1062 :                   param_name_stream__.str(std::string());
1063 :                   param_name_stream__ << "U_transpose" << '.' << j_1__ + 1 << '.' << j_2__
+ 1;
1064 :                   param_names__.push_back(param_name_stream__.str());
1065 :               }
1066 :           }
1067 :           size_t w_j_1_max__ = sum_C;
1068 :           for (size_t j_1__ = 0; j_1__ < w_j_1_max__; ++j_1__) {
1069 :               param_name_stream__.str(std::string());
1070 :               param_name_stream__ << "w" << '.' << j_1__ + 1;
1071 :               param_names__.push_back(param_name_stream__.str());
1072 :           }
1073 :           size_t sum_gamma_j_1_max__ = Q;
1074 :           for (size_t j_1__ = 0; j_1__ < sum_gamma_j_1_max__; ++j_1__) {
1075 :               param_name_stream__.str(std::string());
1076 :               param_name_stream__ << "sum_gamma" << '.' << j_1__ + 1;
1077 :               param_names__.push_back(param_name_stream__.str());
1078 :           }
1079 :       }
1080 :
1081 :       if (!include_gqs__) return;
1082 :   }
1083 :
1084 :
1085 :   void unconstrained_param_names(std::vector<std::string>& param_names__,
1086 :                                  bool include_tparams__ = true,
1087 :                                  bool include_gqs__ = true) const {
```

```
1088 :       std::stringstream param_name_stream__;
1089 :       size_t theta_i1_j_1_max__ = (TT - 1);
1090 :       for (size_t j_1__ = 0; j_1__ < theta_i1_j_1_max__; ++j_1__) {
1091 :         param_name_stream__.str(std::string());
1092 :         param_name_stream__ << "theta_i1" << '.' << j_1__ + 1;
1093 :         param_names__.push_back(param_name_stream__.str());
1094 :       }
1095 :       size_t theta_i2toQ_j_1_max__ = dim_theta_i2toQ;
1096 :       for (size_t j_1__ = 0; j_1__ < theta_i2toQ_j_1_max__; ++j_1__) {
1097 :         param_name_stream__.str(std::string());
1098 :         param_name_stream__ << "theta_i2toQ" << '.' << j_1__ + 1;
1099 :         param_names__.push_back(param_name_stream__.str());
1100 :       }
1101 :       size_t gamma_w_j_1_max__ = (sum_C - Q);
1102 :       for (size_t j_1__ = 0; j_1__ < gamma_w_j_1_max__; ++j_1__) {
1103 :         param_name_stream__.str(std::string());
1104 :         param_name_stream__ << "gamma_w" << '.' << j_1__ + 1;
1105 :         param_names__.push_back(param_name_stream__.str());
1106 :       }
1107 :       size_t mu_free_j_1_max__ = sum_C;
1108 :       for (size_t j_1__ = 0; j_1__ < mu_free_j_1_max__; ++j_1__) {
1109 :         param_name_stream__.str(std::string());
1110 :         param_name_stream__ << "mu_free" << '.' << j_1__ + 1;
1111 :         param_names__.push_back(param_name_stream__.str());
1112 :       }
1113 :       size_t log_sigma_free_sq_j_1_max__ = sum_C;
1114 :       for (size_t j_1__ = 0; j_1__ < log_sigma_free_sq_j_1_max__; ++j_1__) {
1115 :         param_name_stream__.str(std::string());
1116 :         param_name_stream__ << "log_sigma_free_sq" << '.' << j_1__ + 1;
1117 :         param_names__.push_back(param_name_stream__.str());
1118 :       }
1119 :
1120 :       if (!include_gqs__ && !include_tparams__) return;
1121 :
1122 :       if (include_tparams__) {
1123 :         size_t U_transpose_j_2_max__ = TT;
1124 :         size_t U_transpose_j_1_max__ = TT;
1125 :         for (size_t j_2__ = 0; j_2__ < U_transpose_j_2_max__; ++j_2__) {
1126 :           for (size_t j_1__ = 0; j_1__ < U_transpose_j_1_max__; ++j_1__) {
1127 :             param_name_stream__.str(std::string());
1128 :             param_name_stream__ << "U_transpose" << '.' << j_1__ + 1 << '.' << j_2__
+ 1;
1129 :             param_names__.push_back(param_name_stream__.str());
1130 :           }
1131 :         }
1132 :         size_t w_j_1_max__ = sum_C;
1133 :         for (size_t j_1__ = 0; j_1__ < w_j_1_max__; ++j_1__) {
1134 :           param_name_stream__.str(std::string());
1135 :           param_name_stream__ << "w" << '.' << j_1__ + 1;
```

```
1136 :                 param_names__.push_back(param_name_stream__.str());
1137 :             }
1138 :             size_t sum_gamma_j_1_max__ = Q;
1139 :             for (size_t j_1__ = 0; j_1__ < sum_gamma_j_1_max__; ++j_1__) {
1140 :                 param_name_stream__.str(std::string());
1141 :                 param_name_stream__ << "sum_gamma" << '.' << j_1__ + 1;
1142 :                 param_names__.push_back(param_name_stream__.str());
1143 :             }
1144 :         }
1145 :
1146 :         if (!include_gqs__) return;
1147 :     }
1148 :
1149 : }; // model
1150 :
1151 : }  // namespace
1152 :
1153 : typedef
model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model5baf7680f1_BLC
A_unconstrained_Givens_fixC_noZ stan_model;
1154 :
1155 : #include <rstan/rstaninc.hpp>
1156 : /**
1157 :  * Define Rcpp Module to expose stan_fit's functions to R.
1158 :  */
1159 :
RCPP_MODULE(stan_fit4model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_mod){
1160 :
Rcpp::class_<rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_names
pace::model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
1161 :                 boost::random::ecuyer1988>
>("stan_fit4model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ")
1162 :     // .constructor<Rcpp::List>()
1163 :     .constructor<SEXP, SEXP, SEXP>()
1164 :     // .constructor<SEXP, SEXP>()
1165 :     .method("call_sampler",
1166 :
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::call_sampler)
1167 :     .method("param_names",
1168 :
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::param_names)
1169 :     .method("param_names_oi",
1170 :
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
```

5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::param_names_oi)
1171 :   .method("param_fnames_oi",
1172 :
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::param_fnames_oi)
1173 :   .method("param_dims",
1174 :
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::param_dims)
1175 :   .method("param_dims_oi",
1176 :
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::param_dims_oi)
1177 :   .method("update_param_oi",
1178 :
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::update_param_oi)
1179 :   .method("param_oi_tidx",
1180 :
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::param_oi_tidx)
1181 :   .method("grad_log_prob",
1182 :
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::grad_log_prob)
1183 :   .method("log_prob",
1184 :
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ, boost::random::ecuyer1988>::log_prob)
1185 :   .method("unconstrain_pars",
1186 :
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::unconstrain_pars)
1187 :   .method("constrain_pars",
1188 :
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::constrain_pars)
1189 :   .method("num_pars_unconstrained",
1190 :
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model

```
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::num_pars_unconstrained)
1191 :     .method("unconstrained_param_names",
1192 :
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::unconstrained_param_names)
1193 :     .method("constrained_param_names",
1194 :
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::constrained_param_names)
1195 :     .method("standalone_gqs",
1196 :
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::standalone_gqs)
1197 :   ;
1198 : }
1199 :
1200 : // declarations
1201 : extern "C" {
1202 : SEXP file5ba7775a38f( ) ;
1203 : }
1204 :
1205 : // definition
1206 :
1207 : SEXP file5ba7775a38f(  ){
1208 :  return Rcpp::wrap("BLCA_unconstrained_Givens_fixC_noZ");
1209 : }
1210 :
1211 :
Compilation argument:
 /usr/lib64/R/bin/R CMD SHLIB file5ba7775a38f.cpp 2> file5ba7775a38f.cpp.err.txt
In file included from /usr/include/c++/4.8.2/type_traits:35:0,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/prim/scal/meta/is_constant.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/prim/arr/meta/is_constant_struct.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/prim/mat/meta/is_constant_struct.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/prim/mat.hpp:15,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/rev/mat.hpp:12,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/src/
stan/model/model_header.hpp:4,
```

```
                 from file5ba7775a38f.cpp:8:
/usr/include/c++/4.8.2/bits/c++0x_warning.h:32:2: error: #error This file requires compiler and
library support for the ISO C++ 2011 standard. This support is currently experimental, and must
be enabled with the -std=c++11 or -std=gnu++11 compiler options.
 #error This file requires compiler and library support for the \
  ^
In file included from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/
stan/math/rev/core/autodiffstackstorage.hpp:4:0,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/rev/core.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/rev/mat.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/src/
stan/model/model_header.hpp:4,
                 from file5ba7775a38f.cpp:8:
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/memory/
stack_alloc.hpp: In destructor ‘stan::math::stack_alloc::~stack_alloc()’:
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/memory/
stack_alloc.hpp:145:16: error: ISO C++ forbids declaration of ‘block’ with no type [-fpermissive]
    for (auto& block : blocks_)
               ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/memory/
stack_alloc.hpp:145:24: error: range-based ‘for’ loops are not allowed in C++98 mode
    for (auto& block : blocks_)
                       ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/memory/
stack_alloc.hpp:147:19: error: invalid conversion from ‘int’ to ‘void*’ [-fpermissive]
        free(block);
                   ^
In file included from /usr/include/c++/4.8.2/cstdlib:72:0,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/memory/stack_alloc.hpp:8,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/rev/core/autodiffstackstorage.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/rev/core.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/rev/mat.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/src/
stan/model/model_header.hpp:4,
                 from file5ba7775a38f.cpp:8:
/usr/include/stdlib.h:482:13: error:   initializing argument 1 of ‘void free(void*)’ [-fpermissive]
 extern void free (void *__ptr) __THROW;
             ^
```

```
In file included from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/
stan/math/rev/core.hpp:4:0,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/rev/mat.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/src/
stan/model/model_header.hpp:4,
                 from file5ba7775a38f.cpp:8:
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/rev/core/
autodiffstackstorage.hpp: At global scope:
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/rev/core/
autodiffstackstorage.hpp:47:69: warning: defaulted and deleted functions only available with -
std=c++11 or -std=gnu++11 [enabled by default]
     AutodiffStackStorage &operator=(const AutodiffStackStorage &) = delete;
                                                                     ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/rev/core/
autodiffstackstorage.hpp:60:30: warning: defaulted and deleted functions only available with -
std=c++11 or -std=gnu++11 [enabled by default]
   AutodiffStackSingleton() = delete;
                              ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/rev/core/
autodiffstackstorage.hpp:61:71: warning: defaulted and deleted functions only available with -
std=c++11 or -std=gnu++11 [enabled by default]
   explicit AutodiffStackSingleton(AutodiffStackSingleton_t const &) = delete;
                                                                       ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/rev/core/
autodiffstackstorage.hpp:62:73: warning: defaulted and deleted functions only available with -
std=c++11 or -std=gnu++11 [enabled by default]
   AutodiffStackSingleton &operator=(const AutodiffStackSingleton_t &) = delete;
                                                                         ^
In file included from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/
stan/math/rev/core/build_vari_array.hpp:5:0,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/rev/core.hpp:5,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/rev/mat.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/src/
stan/model/model_header.hpp:4,
                 from file5ba7775a38f.cpp:8:
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/rev/core/
var.hpp: In member function ‘bool stan::math::var::is_uninitialized()’:
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/rev/core/
var.hpp:56:63: error: ‘nullptr’ was not declared in this scope
   bool is_uninitialized() { return (vi_ == static_cast<vari*>(nullptr)); }
                                                               ^
```

```
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/rev/core/
var.hpp: In function ‘std::ostream& stan::math::operator<<(std::ostream&, const
stan::math::var&)’:
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/rev/core/
var.hpp:473:18: error: ‘nullptr’ was not declared in this scope
     if (v.vi_ == nullptr)
                  ^
In file included from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/
stan/math/rev/core.hpp:43:0,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/rev/mat.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/src/
stan/model/model_header.hpp:4,
                 from file5ba7775a38f.cpp:8:
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/rev/core/
recover_memory.hpp: In function ‘void stan::math::recover_memory()’:
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/rev/core/
recover_memory.hpp:25:14: error: ISO C++ forbids declaration of ‘x’ with no type [-fpermissive]
   for (auto &x : ChainableStack::instance().var_alloc_stack_) {
              ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/rev/core/
recover_memory.hpp:25:18: error: range-based ‘for’ loops are not allowed in C++98 mode
   for (auto &x : ChainableStack::instance().var_alloc_stack_) {
                  ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/rev/core/
recover_memory.hpp:26:12: error: type ‘int’ argument given to ‘delete’, expected pointer
     delete x;
            ^
In file included from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/
stan/math/rev/core.hpp:45:0,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/rev/mat.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/src/
stan/model/model_header.hpp:4,
                 from file5ba7775a38f.cpp:8:
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/rev/core/
set_zero_all_adjoints.hpp: In function ‘void stan::math::set_zero_all_adjoints()’:
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/rev/core/
set_zero_all_adjoints.hpp:15:14: error: ISO C++ forbids declaration of ‘x’ with no type [-
fpermissive]
   for (auto &x : ChainableStack::instance().var_stack_)
              ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/rev/core/
set_zero_all_adjoints.hpp:15:18: error: range-based ‘for’ loops are not allowed in C++98 mode
   for (auto &x : ChainableStack::instance().var_stack_)
```

```
                ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/rev/core/
set_zero_all_adjoints.hpp:16:6: error: base operand of '->' is not a pointer
     x->set_zero_adjoint();
      ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/rev/core/
set_zero_all_adjoints.hpp:17:14: error: ISO C++ forbids declaration of 'x' with no type [-
fpermissive]
   for (auto &x : ChainableStack::instance().var_nochain_stack_)
              ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/rev/core/
set_zero_all_adjoints.hpp:17:18: error: range-based 'for' loops are not allowed in C++98 mode
   for (auto &x : ChainableStack::instance().var_nochain_stack_)
                  ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/rev/core/
set_zero_all_adjoints.hpp:18:6: error: base operand of '->' is not a pointer
     x->set_zero_adjoint();
      ^
In file included from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/
stan/math/prim/scal/meta/operands_and_partials.hpp:5:0,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/rev/scal/meta/operands_and_partials.hpp:9,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/rev/mat/meta/operands_and_partials.hpp:7,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/rev/mat.hpp:7,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/src/
stan/model/model_header.hpp:4,
                 from file5ba7775a38f.cpp:8:
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/scal/
meta/return_type.hpp: At global scope:
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/scal/
meta/return_type.hpp:32:31: warning: variadic templates only available with -std=c++11 or -
std=gnu++11 [enabled by default]
 template <typename T, typename... Types_pack>
                               ^
In file included from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/
stan/math/rev/mat/vectorize/apply_scalar_unary.hpp:4:0,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/rev/mat.hpp:11,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/src/
stan/model/model_header.hpp:4,
                 from file5ba7775a38f.cpp:8:
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
vectorize/apply_scalar_unary.hpp: In static member function 'static
```

```
stan::math::apply_scalar_unary<F, T>::return_t stan::math::apply_scalar_unary<F,
T>::apply(const T&)':
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
vectorize/apply_scalar_unary.hpp:58:76: warning: lambda expressions only available with -
std=c++11 or -std=gnu++11 [enabled by default]
       [](scalar_t x) { return apply_scalar_unary<F, scalar_t>::apply(x); });
                                                                            ^
In file included from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/
stan/math/prim/mat.hpp:9:0,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/rev/mat.hpp:12,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/src/
stan/model/model_header.hpp:4,
                 from file5ba7775a38f.cpp:8:
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_array_or_scalar.hpp: At global scope:
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_array_or_scalar.hpp:21:36: error: a function call cannot appear in a constant-
expression
    const Eigen::Matrix<T, R, C>& v) {
                                   ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_array_or_scalar.hpp:21:36: error: template argument 3 is invalid
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_array_or_scalar.hpp:21:36: error: template argument 4 is invalid
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_array_or_scalar.hpp:21:36: error: template argument 6 is invalid
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_array_or_scalar.hpp:20:41: error: template argument 1 is invalid
 inline Eigen::ArrayWrapper<const Eigen::Matrix<T, R, C>> as_array_or_scalar(
                                         ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_array_or_scalar.hpp:21:38: error: expected unqualified-id before '{' token
    const Eigen::Matrix<T, R, C>& v) {
                                   ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_array_or_scalar.hpp:35:74: error: template argument 3 is invalid
 inline Eigen::ArrayWrapper<const Eigen::Map<const Eigen::Matrix<T, R, C>>>
                                                                          ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_array_or_scalar.hpp:35:74: error: template argument 4 is invalid
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_array_or_scalar.hpp:35:74: error: template argument 6 is invalid
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_array_or_scalar.hpp:36:69: error: template argument 1 is invalid
 as_array_or_scalar(const Eigen::Map<const Eigen::Matrix<T, R, C>>& v) {
                                                                     ^
```

```
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_array_or_scalar.hpp:35:41: error: template argument 1 is invalid
 inline Eigen::ArrayWrapper<const Eigen::Map<const Eigen::Matrix<T, R, C>>>
                                         ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_array_or_scalar.hpp:36:71: error: expected unqualified-id before '{' token
 as_array_or_scalar(const Eigen::Map<const Eigen::Matrix<T, R, C>>& v) {
                                                                       ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_array_or_scalar.hpp:49:28: error: a function call cannot appear in a constant-
expression
    const std::vector<T>& v) {
                          ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_array_or_scalar.hpp:49:28: error: template argument 3 is invalid
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_array_or_scalar.hpp:49:28: error: template argument 4 is invalid
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_array_or_scalar.hpp:49:28: error: template argument 6 is invalid
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_array_or_scalar.hpp:48:32: error: template argument 1 is invalid
 inline Eigen::Map<const Eigen::Array<T, Eigen::Dynamic, 1>> as_array_or_scalar(
                               ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_array_or_scalar.hpp:49:30: error: expected unqualified-id before '{' token
    const std::vector<T>& v) {
                            ^
In file included from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/
stan/math/prim/mat.hpp:10:0,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/rev/mat.hpp:12,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/src/
stan/model/model_header.hpp:4,
                 from file5ba7775a38f.cpp:8:
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_column_vector_or_scalar.hpp:34:1: error:
'stan::math::as_column_vector_or_scalar(const Eigen::Matrix<Scalar, -1, 1>&)' cannot appear in
a constant-expression
 as_column_vector_or_scalar(const Eigen::Matrix<T, 1, Eigen::Dynamic>& a) {
 ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_column_vector_or_scalar.hpp:34:72: error: a function call cannot appear in a constant-
expression
 as_column_vector_or_scalar(const Eigen::Matrix<T, 1, Eigen::Dynamic>& a) {
                                                                        ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_column_vector_or_scalar.hpp:34:72: error: template argument 3 is invalid
```

```
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_column_vector_or_scalar.hpp:34:72: error: template argument 4 is invalid
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_column_vector_or_scalar.hpp:34:72: error: template argument 6 is invalid
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_column_vector_or_scalar.hpp:33:32: error: template argument 1 is invalid
 inline Eigen::Map<const Eigen::Matrix<T, Eigen::Dynamic, 1>>
                                ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_column_vector_or_scalar.hpp:34:74: error: expected unqualified-id before '{' token
 as_column_vector_or_scalar(const Eigen::Matrix<T, 1, Eigen::Dynamic>& a) {
                                                                          ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_column_vector_or_scalar.hpp:50:1: error:
'stan::math::as_column_vector_or_scalar(const Eigen::Matrix<Scalar, -1, 1>&)' cannot appear in
a constant-expression
 as_column_vector_or_scalar(const std::vector<T>& a) {
 ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_column_vector_or_scalar.hpp:50:51: error: a function call cannot appear in a constant-
expression
 as_column_vector_or_scalar(const std::vector<T>& a) {
                                                   ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_column_vector_or_scalar.hpp:50:51: error: template argument 3 is invalid
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_column_vector_or_scalar.hpp:50:51: error: template argument 4 is invalid
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_column_vector_or_scalar.hpp:50:51: error: template argument 6 is invalid
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_column_vector_or_scalar.hpp:49:32: error: template argument 1 is invalid
 inline Eigen::Map<const Eigen::Matrix<T, Eigen::Dynamic, 1>>
                                ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/
meta/as_column_vector_or_scalar.hpp:50:53: error: expected unqualified-id before '{' token
 as_column_vector_or_scalar(const std::vector<T>& a) {
                                                     ^
In file included from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/
stan/math/prim/arr/meta/is_constant_struct.hpp:4:0,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/prim/mat/meta/is_constant_struct.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/prim/mat.hpp:15,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/rev/mat.hpp:12,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/src/
stan/model/model_header.hpp:4,
```

```
                 from file5ba7775a38f.cpp:8:
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/scal/
meta/is_constant.hpp:27:18: error: ‘is_convertible’ is not a member of ‘std’
   enum { value = std::is_convertible<T, double>::value };
                  ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/scal/
meta/is_constant.hpp:27:18: note: suggested alternatives:
In file included from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/RcppEigen/include/
Eigen/Core:364:0,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/RcppEigen/include/Eigen/
Dense:1,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/prim/mat/fun/Eigen.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/rev/core/build_vari_array.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/rev/core.hpp:5,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/rev/mat.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/src/
stan/model/model_header.hpp:4,
                 from file5ba7775a38f.cpp:8:
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/RcppEigen/include/Eigen/src/Core/util/
Meta.h:153:8: note:   ‘Eigen::internal::is_convertible’
 struct is_convertible
        ^
In file included from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/BH/include/boost/
type_traits/is_constructible.hpp:70:0,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/BH/include/boost/math/
tools/convert_from_string.hpp:13,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/BH/include/boost/math/
constants/constants.hpp:13,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/prim/scal/fun/constants.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/rev/core/operator_unary_plus.hpp:7,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/rev/core.hpp:37,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/rev/mat.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/src/
stan/model/model_header.hpp:4,
                 from file5ba7775a38f.cpp:8:
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/BH/include/boost/type_traits/
is_convertible.hpp:481:8: note:   ‘boost::is_convertible’
```

```
 struct is_convertible : public
integral_constant<bool, ::boost::detail::is_convertible_impl_dispatch<From, To>::value>
       ^
In file included from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/
stan/math/prim/arr/meta/is_constant_struct.hpp:4:0,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/prim/mat/meta/is_constant_struct.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/prim/mat.hpp:15,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math/rev/mat.hpp:12,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/
math.hpp:4,
                 from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/src/
stan/model/model_header.hpp:4,
                 from file5ba7775a38f.cpp:8:
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/scal/
meta/is_constant.hpp:27:39: error: expected primary-expression before ',' token
   enum { value = std::is_convertible<T, double>::value };
                                       ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/scal/
meta/is_constant.hpp:27:41: error: expected identifier before 'double'
   enum { value = std::is_convertible<T, double>::value };
                                         ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/scal/
meta/is_constant.hpp:27:41: error: expected '}' before 'double'
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/scal/
meta/is_constant.hpp:27:47: error: expected unqualified-id before '>' token
   enum { value = std::is_convertible<T, double>::value };
                                               ^
/home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/scal/
meta/is_constant.hpp:30:1: error: expected declaration before '}' token
 }  // namespace stan
 ^
make: *** [file5ba7775a38f.o] Error 1

ERROR(s) during compilation: source code errors or compiler configuration errors!

Program source:
  1:
  2: // includes from the plugin
  3: // [[Rcpp::plugins(cpp14)]]
  4:
  5: // user includes
  6: #define STAN__SERVICES__COMMAND_HPP// Code generated by Stan version 2.19.1
  7:
  8: #include <stan/model/model_header.hpp>
  9:
 10: namespace model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace {
```

```
 11:
 12: using std::istream;
 13: using std::string;
 14: using std::stringstream;
 15: using std::vector;
 16: using stan::io::dump;
 17: using stan::math::lgamma;
 18: using stan::model::prob_grad;
 19: using namespace stan::math;
 20:
 21: static int current_statement_begin__;
 22:
 23: stan::io::program_reader prog_reader__() {
 24:     stan::io::program_reader reader;
 25:     reader.add_event(0, 0, "start",
"model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ");
 26:     reader.add_event(164, 162, "end",
"model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ");
 27:     return reader;
 28: }
 29:
 30: class model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ : public prob_grad {
 31: private:
 32:         int TT;
 33:         int Q;
 34:         std::vector<int> C;
 35:         int V;
 36:         matrix_d X;
 37:         double delta_w;
 38:         double prior_mean_mu_free;
 39:         double prior_precision_mu_free;
 40:         double prior_mean_log_sigma_free_sq;
 41:         double prior_sd_log_sigma_free_sq;
 42:         double prior_sd_mu_free;
 43:         int dim_theta_i2toQ;
 44:         int sum_C;
 45:         std::vector<int> start_pos_gamma;
 46:         std::vector<int> start_pos_GMMpara;
 47: public:
 48:     model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ(stan::io::var_context&
context__,
 49:         std::ostream* pstream__ = 0)
 50:         : prob_grad(0) {
 51:         ctor_body(context__, 0, pstream__);
 52:     }
 53:
 54:     model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ(stan::io::var_context&
context__,
 55:         unsigned int random_seed__,
```

```
 56:         std::ostream* pstream__ = 0)
 57:         : prob_grad(0) {
 58:         ctor_body(context__, random_seed__, pstream__);
 59:     }
 60:
 61:     void ctor_body(stan::io::var_context& context__,
 62:                    unsigned int random_seed__,
 63:                    std::ostream* pstream__) {
 64:         typedef double local_scalar_t__;
 65:
 66:         boost::ecuyer1988 base_rng__ =
 67:           stan::services::util::create_rng(random_seed__, 0);
 68:         (void) base_rng__;  // suppress unused var warning
 69:
 70:         current_statement_begin__ = -1;
 71:
 72:         static const char* function__ =
"model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model5baf7680f1_BL
CA_unconstrained_Givens_fixC_noZ";
 73:         (void) function__;  // dummy to suppress unused var warning
 74:         size_t pos__;
 75:         (void) pos__;  // dummy to suppress unused var warning
 76:         std::vector<int> vals_i__;
 77:         std::vector<double> vals_r__;
 78:         local_scalar_t__ DUMMY_VAR__(std::numeric_limits<double>::quiet_NaN());
 79:         (void) DUMMY_VAR__;  // suppress unused var warning
 80:
 81:         try {
 82:             // initialize data block variables from context__
 83:             current_statement_begin__ = 2;
 84:             context__.validate_dims("data initialization", "TT", "int", context__.to_vec());
 85:             TT = int(0);
 86:             vals_i__ = context__.vals_i("TT");
 87:             pos__ = 0;
 88:             TT = vals_i__[pos__++];
 89:             check_greater_or_equal(function__, "TT", TT, 1);
 90:
 91:             current_statement_begin__ = 3;
 92:             context__.validate_dims("data initialization", "Q", "int", context__.to_vec());
 93:             Q = int(0);
 94:             vals_i__ = context__.vals_i("Q");
 95:             pos__ = 0;
 96:             Q = vals_i__[pos__++];
 97:             check_greater_or_equal(function__, "Q", Q, 1);
 98:             check_less_or_equal(function__, "Q", Q, TT);
 99:
100:             current_statement_begin__ = 4;
101:             validate_non_negative_index("C", "Q", Q);
102:             context__.validate_dims("data initialization", "C", "int", context__.to_vec(Q));
```

```
103:             C = std::vector<int>(Q, int(0));
104:             vals_i__ = context__.vals_i("C");
105:             pos__ = 0;
106:             size_t C_k_0_max__ = Q;
107:             for (size_t k_0__ = 0; k_0__ < C_k_0_max__; ++k_0__) {
108:                 C[k_0__] = vals_i__[pos__++];
109:             }
110:             size_t C_i_0_max__ = Q;
111:             for (size_t i_0__ = 0; i_0__ < C_i_0_max__; ++i_0__) {
112:                 check_greater_or_equal(function__, "C[i_0__]", C[i_0__], 2);
113:             }
114:
115:             current_statement_begin__ = 5;
116:             context__.validate_dims("data initialization", "V", "int", context__.to_vec());
117:             V = int(0);
118:             vals_i__ = context__.vals_i("V");
119:             pos__ = 0;
120:             V = vals_i__[pos__++];
121:             check_greater_or_equal(function__, "V", V, 1);
122:
123:             current_statement_begin__ = 6;
124:             validate_non_negative_index("X", "V", V);
125:             validate_non_negative_index("X", "TT", TT);
126:             context__.validate_dims("data initialization", "X", "matrix_d",
context__.to_vec(V,TT));
127:             X = Eigen::Matrix<double, Eigen::Dynamic, Eigen::Dynamic>(V, TT);
128:             vals_r__ = context__.vals_r("X");
129:             pos__ = 0;
130:             size_t X_j_2_max__ = TT;
131:             size_t X_j_1_max__ = V;
132:             for (size_t j_2__ = 0; j_2__ < X_j_2_max__; ++j_2__) {
133:                 for (size_t j_1__ = 0; j_1__ < X_j_1_max__; ++j_1__) {
134:                     X(j_1__, j_2__) = vals_r__[pos__++];
135:                 }
136:             }
137:
138:             current_statement_begin__ = 8;
139:             context__.validate_dims("data initialization", "delta_w", "double",
context__.to_vec());
140:             delta_w = double(0);
141:             vals_r__ = context__.vals_r("delta_w");
142:             pos__ = 0;
143:             delta_w = vals_r__[pos__++];
144:             check_greater_or_equal(function__, "delta_w", delta_w, 0);
145:
146:             current_statement_begin__ = 9;
147:             context__.validate_dims("data initialization", "prior_mean_mu_free", "double",
context__.to_vec());
148:             prior_mean_mu_free = double(0);
```

```
149:             vals_r__ = context__.vals_r("prior_mean_mu_free");
150:             pos__ = 0;
151:             prior_mean_mu_free = vals_r__[pos__++];
152:
153:             current_statement_begin__ = 10;
154:             context__.validate_dims("data initialization", "prior_precision_mu_free", "double",
context__.to_vec());
155:             prior_precision_mu_free = double(0);
156:             vals_r__ = context__.vals_r("prior_precision_mu_free");
157:             pos__ = 0;
158:             prior_precision_mu_free = vals_r__[pos__++];
159:             check_greater_or_equal(function__, "prior_precision_mu_free",
prior_precision_mu_free, 0);
160:
161:             current_statement_begin__ = 11;
162:             context__.validate_dims("data initialization", "prior_mean_log_sigma_free_sq",
"double", context__.to_vec());
163:             prior_mean_log_sigma_free_sq = double(0);
164:             vals_r__ = context__.vals_r("prior_mean_log_sigma_free_sq");
165:             pos__ = 0;
166:             prior_mean_log_sigma_free_sq = vals_r__[pos__++];
167:
168:             current_statement_begin__ = 12;
169:             context__.validate_dims("data initialization", "prior_sd_log_sigma_free_sq",
"double", context__.to_vec());
170:             prior_sd_log_sigma_free_sq = double(0);
171:             vals_r__ = context__.vals_r("prior_sd_log_sigma_free_sq");
172:             pos__ = 0;
173:             prior_sd_log_sigma_free_sq = vals_r__[pos__++];
174:             check_greater_or_equal(function__, "prior_sd_log_sigma_free_sq",
prior_sd_log_sigma_free_sq, 0);
175:
176:
177:             // initialize transformed data variables
178:             current_statement_begin__ = 16;
179:             prior_sd_mu_free = double(0);
180:             stan::math::fill(prior_sd_mu_free, DUMMY_VAR__);
181:             stan::math::assign(prior_sd_mu_free,(1 /
stan::math::sqrt(prior_precision_mu_free)));
182:
183:             current_statement_begin__ = 17;
184:             dim_theta_i2toQ = int(0);
185:             stan::math::fill(dim_theta_i2toQ, std::numeric_limits<int>::min());
186:             stan::math::assign(dim_theta_i2toQ,((TT * (Q - 1)) - divide(((Q - 1) * (Q + 2)), 2)));
187:
188:             current_statement_begin__ = 18;
189:             sum_C = int(0);
190:             stan::math::fill(sum_C, std::numeric_limits<int>::min());
191:
```

```
192:         current_statement_begin__ = 19;
193:         validate_non_negative_index("start_pos_gamma", "(Q + 1)", (Q + 1));
194:         start_pos_gamma = std::vector<int>((Q + 1), int(0));
195:         stan::math::fill(start_pos_gamma, std::numeric_limits<int>::min());
196:
197:         current_statement_begin__ = 20;
198:         validate_non_negative_index("start_pos_GMMpara", "(Q + 1)", (Q + 1));
199:         start_pos_GMMpara = std::vector<int>((Q + 1), int(0));
200:         stan::math::fill(start_pos_GMMpara, std::numeric_limits<int>::min());
201:
202:         // execute transformed data statements
203:         current_statement_begin__ = 21;
204:         stan::model::assign(start_pos_gamma,
205:                   stan::model::cons_list(stan::model::index_uni(1),
stan::model::nil_index_list()),
206:                   1,
207:                   "assigning variable start_pos_gamma");
208:         current_statement_begin__ = 22;
209:         stan::model::assign(start_pos_GMMpara,
210:                   stan::model::cons_list(stan::model::index_uni(1),
stan::model::nil_index_list()),
211:                   1,
212:                   "assigning variable start_pos_GMMpara");
213:         current_statement_begin__ = 23;
214:         if (as_bool(logical_gt(Q, 1))) {
215:
216:           current_statement_begin__ = 24;
217:           for (int k = 1; k <= Q; ++k) {
218:
219:             current_statement_begin__ = 25;
220:             stan::model::assign(start_pos_gamma,
221:                       stan::model::cons_list(stan::model::index_uni((k + 1)),
stan::model::nil_index_list()),
222:                       ((get_base1(start_pos_gamma, k, "start_pos_gamma", 1) +
get_base1(C, k, "C", 1)) - 1),
223:                       "assigning variable start_pos_gamma");
224:             current_statement_begin__ = 26;
225:             stan::model::assign(start_pos_GMMpara,
226:                       stan::model::cons_list(stan::model::index_uni((k + 1)),
stan::model::nil_index_list()),
227:                       (get_base1(start_pos_GMMpara, k, "start_pos_GMMpara", 1) +
get_base1(C, k, "C", 1)),
228:                       "assigning variable start_pos_GMMpara");
229:           }
230:         }
231:         current_statement_begin__ = 29;
232:         stan::math::assign(sum_C, (get_base1(start_pos_GMMpara, (Q + 1),
"start_pos_GMMpara", 1) - 1));
233:
```

```
234:             // validate transformed data
235:
236:             // validate, set parameter ranges
237:             num_params_r__ = 0U;
238:             param_ranges_i__.clear();
239:             current_statement_begin__ = 32;
240:             validate_non_negative_index("theta_i1", "(TT - 1)", (TT - 1));
241:             num_params_r__ += (TT - 1);
242:             current_statement_begin__ = 33;
243:             validate_non_negative_index("theta_i2toQ", "dim_theta_i2toQ", dim_theta_i2toQ);
244:             num_params_r__ += dim_theta_i2toQ;
245:             current_statement_begin__ = 34;
246:             validate_non_negative_index("gamma_w", "(sum_C - Q)", (sum_C - Q));
247:             num_params_r__ += (sum_C - Q);
248:             current_statement_begin__ = 35;
249:             validate_non_negative_index("mu_free", "sum_C", sum_C);
250:             num_params_r__ += sum_C;
251:             current_statement_begin__ = 36;
252:             validate_non_negative_index("log_sigma_free_sq", "sum_C", sum_C);
253:             num_params_r__ += sum_C;
254:         } catch (const std::exception& e) {
255:             stan::lang::rethrow_located(e, current_statement_begin__, prog_reader__());
256:             // Next line prevents compiler griping about no return
257:             throw std::runtime_error("*** IF YOU SEE THIS, PLEASE REPORT A BUG ***");
258:         }
259:     }
260:
261:     ~model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ() { }
262:
263:
264:     void transform_inits(const stan::io::var_context& context__,
265:                          std::vector<int>& params_i__,
266:                          std::vector<double>& params_r__,
267:                          std::ostream* pstream__) const {
268:         typedef double local_scalar_t__;
269:         stan::io::writer<double> writer__(params_r__, params_i__);
270:         size_t pos__;
271:         (void) pos__; // dummy call to supress warning
272:         std::vector<double> vals_r__;
273:         std::vector<int> vals_i__;
274:
275:         current_statement_begin__ = 32;
276:         if (!(context__.contains_r("theta_i1")))
277:             stan::lang::rethrow_located(std::runtime_error(std::string("Variable theta_i1
missing")), current_statement_begin__, prog_reader__());
278:         vals_r__ = context__.vals_r("theta_i1");
279:         pos__ = 0U;
280:         validate_non_negative_index("theta_i1", "(TT - 1)", (TT - 1));
```

```
281:         context__.validate_dims("parameter initialization", "theta_i1", "vector_d",
context__.to_vec((TT - 1)));
282:         Eigen::Matrix<double, Eigen::Dynamic, 1> theta_i1((TT - 1));
283:         size_t theta_i1_j_1_max__ = (TT - 1);
284:         for (size_t j_1__ = 0; j_1__ < theta_i1_j_1_max__; ++j_1__) {
285:             theta_i1(j_1__) = vals_r__[pos__++];
286:         }
287:         try {
288:             writer__.vector_lub_unconstrain(-(stan::math::pi()), stan::math::pi(), theta_i1);
289:         } catch (const std::exception& e) {
290:             stan::lang::rethrow_located(std::runtime_error(std::string("Error transforming
variable theta_i1: ") + e.what()), current_statement_begin__, prog_reader__());
291:         }
292:
293:         current_statement_begin__ = 33;
294:         if (!(context__.contains_r("theta_i2toQ")))
295:             stan::lang::rethrow_located(std::runtime_error(std::string("Variable theta_i2toQ
missing")), current_statement_begin__, prog_reader__());
296:         vals_r__ = context__.vals_r("theta_i2toQ");
297:         pos__ = 0U;
298:         validate_non_negative_index("theta_i2toQ", "dim_theta_i2toQ", dim_theta_i2toQ);
299:         context__.validate_dims("parameter initialization", "theta_i2toQ", "vector_d",
context__.to_vec(dim_theta_i2toQ));
300:         Eigen::Matrix<double, Eigen::Dynamic, 1> theta_i2toQ(dim_theta_i2toQ);
301:         size_t theta_i2toQ_j_1_max__ = dim_theta_i2toQ;
302:         for (size_t j_1__ = 0; j_1__ < theta_i2toQ_j_1_max__; ++j_1__) {
303:             theta_i2toQ(j_1__) = vals_r__[pos__++];
304:         }
305:         try {
306:             writer__.vector_lub_unconstrain(0, stan::math::pi(), theta_i2toQ);
307:         } catch (const std::exception& e) {
308:             stan::lang::rethrow_located(std::runtime_error(std::string("Error transforming
variable theta_i2toQ: ") + e.what()), current_statement_begin__, prog_reader__());
309:         }
310:
311:         current_statement_begin__ = 34;
312:         if (!(context__.contains_r("gamma_w")))
313:             stan::lang::rethrow_located(std::runtime_error(std::string("Variable gamma_w
missing")), current_statement_begin__, prog_reader__());
314:         vals_r__ = context__.vals_r("gamma_w");
315:         pos__ = 0U;
316:         validate_non_negative_index("gamma_w", "(sum_C - Q)", (sum_C - Q));
317:         context__.validate_dims("parameter initialization", "gamma_w", "vector_d",
context__.to_vec((sum_C - Q)));
318:         Eigen::Matrix<double, Eigen::Dynamic, 1> gamma_w((sum_C - Q));
319:         size_t gamma_w_j_1_max__ = (sum_C - Q);
320:         for (size_t j_1__ = 0; j_1__ < gamma_w_j_1_max__; ++j_1__) {
321:             gamma_w(j_1__) = vals_r__[pos__++];
322:         }
```

```
323:         try {
324:             writer__.vector_lb_unconstrain(0, gamma_w);
325:         } catch (const std::exception& e) {
326:             stan::lang::rethrow_located(std::runtime_error(std::string("Error transforming
variable gamma_w: ") + e.what()), current_statement_begin__, prog_reader__());
327:         }
328:
329:         current_statement_begin__ = 35;
330:         if (!(context__.contains_r("mu_free")))
331:             stan::lang::rethrow_located(std::runtime_error(std::string("Variable mu_free
missing")), current_statement_begin__, prog_reader__());
332:         vals_r__ = context__.vals_r("mu_free");
333:         pos__ = 0U;
334:         validate_non_negative_index("mu_free", "sum_C", sum_C);
335:         context__.validate_dims("parameter initialization", "mu_free", "vector_d",
context__.to_vec(sum_C));
336:         Eigen::Matrix<double, Eigen::Dynamic, 1> mu_free(sum_C);
337:         size_t mu_free_j_1_max__ = sum_C;
338:         for (size_t j_1__ = 0; j_1__ < mu_free_j_1_max__; ++j_1__) {
339:             mu_free(j_1__) = vals_r__[pos__++];
340:         }
341:         try {
342:             writer__.vector_unconstrain(mu_free);
343:         } catch (const std::exception& e) {
344:             stan::lang::rethrow_located(std::runtime_error(std::string("Error transforming
variable mu_free: ") + e.what()), current_statement_begin__, prog_reader__());
345:         }
346:
347:         current_statement_begin__ = 36;
348:         if (!(context__.contains_r("log_sigma_free_sq")))
349:             stan::lang::rethrow_located(std::runtime_error(std::string("Variable
log_sigma_free_sq missing")), current_statement_begin__, prog_reader__());
350:         vals_r__ = context__.vals_r("log_sigma_free_sq");
351:         pos__ = 0U;
352:         validate_non_negative_index("log_sigma_free_sq", "sum_C", sum_C);
353:         context__.validate_dims("parameter initialization", "log_sigma_free_sq", "vector_d",
context__.to_vec(sum_C));
354:         Eigen::Matrix<double, Eigen::Dynamic, 1> log_sigma_free_sq(sum_C);
355:         size_t log_sigma_free_sq_j_1_max__ = sum_C;
356:         for (size_t j_1__ = 0; j_1__ < log_sigma_free_sq_j_1_max__; ++j_1__) {
357:             log_sigma_free_sq(j_1__) = vals_r__[pos__++];
358:         }
359:         try {
360:             writer__.vector_unconstrain(log_sigma_free_sq);
361:         } catch (const std::exception& e) {
362:             stan::lang::rethrow_located(std::runtime_error(std::string("Error transforming
variable log_sigma_free_sq: ") + e.what()), current_statement_begin__, prog_reader__());
363:         }
364:
```

```
365:       params_r__ = writer__.data_r();
366:       params_i__ = writer__.data_i();
367:     }
368:
369:     void transform_inits(const stan::io::var_context& context,
370:                          Eigen::Matrix<double, Eigen::Dynamic, 1>& params_r,
371:                          std::ostream* pstream__) const {
372:       std::vector<double> params_r_vec;
373:       std::vector<int> params_i_vec;
374:       transform_inits(context, params_i_vec, params_r_vec, pstream__);
375:       params_r.resize(params_r_vec.size());
376:       for (int i = 0; i < params_r.size(); ++i)
377:         params_r(i) = params_r_vec[i];
378:     }
379:
380:
381:     template <bool propto__, bool jacobian__, typename T__>
382:     T__ log_prob(std::vector<T__>& params_r__,
383:                  std::vector<int>& params_i__,
384:                  std::ostream* pstream__ = 0) const {
385:
386:       typedef T__ local_scalar_t__;
387:
388:       local_scalar_t__ DUMMY_VAR__(std::numeric_limits<double>::quiet_NaN());
389:       (void) DUMMY_VAR__;  // dummy to suppress unused var warning
390:
391:       T__ lp__(0.0);
392:       stan::math::accumulator<T__> lp_accum__;
393:       try {
394:           stan::io::reader<local_scalar_t__> in__(params_r__, params_i__);
395:
396:           // model parameters
397:           current_statement_begin__ = 32;
398:           Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> theta_i1;
399:           (void) theta_i1;  // dummy to suppress unused var warning
400:           if (jacobian__)
401:               theta_i1 = in__.vector_lub_constrain(-(stan::math::pi()), stan::math::pi(), (TT - 1),
lp__);
402:           else
403:               theta_i1 = in__.vector_lub_constrain(-(stan::math::pi()), stan::math::pi(), (TT - 1));
404:
405:           current_statement_begin__ = 33;
406:           Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> theta_i2toQ;
407:           (void) theta_i2toQ;  // dummy to suppress unused var warning
408:           if (jacobian__)
409:               theta_i2toQ = in__.vector_lub_constrain(0, stan::math::pi(), dim_theta_i2toQ,
lp__);
410:           else
411:               theta_i2toQ = in__.vector_lub_constrain(0, stan::math::pi(), dim_theta_i2toQ);
```

```
412:
413:         current_statement_begin__ = 34;
414:         Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> gamma_w;
415:         (void) gamma_w;  // dummy to suppress unused var warning
416:         if (jacobian__)
417:           gamma_w = in__.vector_lb_constrain(0, (sum_C - Q), lp__);
418:         else
419:           gamma_w = in__.vector_lb_constrain(0, (sum_C - Q));
420:
421:         current_statement_begin__ = 35;
422:         Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> mu_free;
423:         (void) mu_free;  // dummy to suppress unused var warning
424:         if (jacobian__)
425:           mu_free = in__.vector_constrain(sum_C, lp__);
426:         else
427:           mu_free = in__.vector_constrain(sum_C);
428:
429:         current_statement_begin__ = 36;
430:         Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> log_sigma_free_sq;
431:         (void) log_sigma_free_sq;  // dummy to suppress unused var warning
432:         if (jacobian__)
433:           log_sigma_free_sq = in__.vector_constrain(sum_C, lp__);
434:         else
435:           log_sigma_free_sq = in__.vector_constrain(sum_C);
436:
437:         // transformed parameters
438:         current_statement_begin__ = 39;
439:         validate_non_negative_index("U_transpose", "TT", TT);
440:         validate_non_negative_index("U_transpose", "TT", TT);
441:         Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, Eigen::Dynamic>
U_transpose(TT, TT);
442:         stan::math::initialize(U_transpose, DUMMY_VAR__);
443:         stan::math::fill(U_transpose, DUMMY_VAR__);
444:         stan::math::assign(U_transpose,diag_matrix(rep_vector(1, TT)));
445:
446:         current_statement_begin__ = 40;
447:         validate_non_negative_index("w", "sum_C", sum_C);
448:         Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> w(sum_C);
449:         stan::math::initialize(w, DUMMY_VAR__);
450:         stan::math::fill(w, DUMMY_VAR__);
451:
452:         current_statement_begin__ = 44;
453:         validate_non_negative_index("sum_gamma", "Q", Q);
454:         Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> sum_gamma(Q);
455:         stan::math::initialize(sum_gamma, DUMMY_VAR__);
456:         stan::math::fill(sum_gamma, DUMMY_VAR__);
457:
458:         // transformed parameters block statements
459:         {
```

```
460:          current_statement_begin__ = 49;
461:          int index(0);
462:          (void) index;  // dummy to suppress unused var warning
463:          stan::math::fill(index, std::numeric_limits<int>::min());
464:          stan::math::assign(index,1);
465:
466:          current_statement_begin__ = 50;
467:          local_scalar_t__ cos1(DUMMY_VAR__);
468:          (void) cos1;  // dummy to suppress unused var warning
469:          stan::math::initialize(cos1, DUMMY_VAR__);
470:          stan::math::fill(cos1, DUMMY_VAR__);
471:
472:          current_statement_begin__ = 51;
473:          local_scalar_t__ sin1(DUMMY_VAR__);
474:          (void) sin1;  // dummy to suppress unused var warning
475:          stan::math::initialize(sin1, DUMMY_VAR__);
476:          stan::math::fill(sin1, DUMMY_VAR__);
477:
478:          current_statement_begin__ = 52;
479:          validate_non_negative_index("Ui_tmp", "TT", TT);
480:          Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> Ui_tmp(TT);
481:          stan::math::initialize(Ui_tmp, DUMMY_VAR__);
482:          stan::math::fill(Ui_tmp, DUMMY_VAR__);
483:
484:          current_statement_begin__ = 53;
485:          validate_non_negative_index("Uj_tmp", "TT", TT);
486:          Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> Uj_tmp(TT);
487:          stan::math::initialize(Uj_tmp, DUMMY_VAR__);
488:          stan::math::fill(Uj_tmp, DUMMY_VAR__);
489:
490:
491:          current_statement_begin__ = 55;
492:          for (int k = 1; k <= Q; ++k) {
493:            {
494:              current_statement_begin__ = 56;
495:              validate_non_negative_index("gamma_w_seg", "(get_base1(C, k, \"C\", 1) - 1)",
(get_base1(C, k, "C", 1) - 1));
496:              Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1>
gamma_w_seg((get_base1(C, k, "C", 1) - 1));
497:              stan::math::initialize(gamma_w_seg, DUMMY_VAR__);
498:              stan::math::fill(gamma_w_seg, DUMMY_VAR__);
499:              stan::math::assign(gamma_w_seg,segment(gamma_w,
get_base1(start_pos_gamma, k, "start_pos_gamma", 1), (get_base1(C, k, "C", 1) - 1)));
500:
501:
502:              current_statement_begin__ = 60;
503:              if (as_bool(logical_eq(k, 1))) {
504:
505:                current_statement_begin__ = 61;
```

```
506:                 for (int j = (k + 1); j <= TT; ++j) {
507:
508:                     current_statement_begin__ = 62;
509:                     stan::math::assign(cos1, stan::math::cos(get_base1(theta_i1, index,
"theta_i1", 1)));
510:                     current_statement_begin__ = 63;
511:                     stan::math::assign(sin1, stan::math::sin(get_base1(theta_i1, index,
"theta_i1", 1)));
512:                     current_statement_begin__ = 64;
513:                     stan::math::assign(Ui_tmp, subtract(multiply(cos1,
stan::model::rvalue(U_transpose, stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(k), stan::model::nil_index_list())), "U_transpose")),
multiply(sin1, stan::model::rvalue(U_transpose,
stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(j), stan::model::nil_index_list())),
"U_transpose"))));
514:                     current_statement_begin__ = 65;
515:                     stan::math::assign(Uj_tmp, add(multiply(sin1,
stan::model::rvalue(U_transpose, stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(k), stan::model::nil_index_list())), "U_transpose")),
multiply(cos1, stan::model::rvalue(U_transpose,
stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(j), stan::model::nil_index_list())),
"U_transpose"))));
516:                     current_statement_begin__ = 66;
517:                     stan::model::assign(U_transpose,
518:                             stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(k), stan::model::nil_index_list())),
519:                             Ui_tmp,
520:                             "assigning variable U_transpose");
521:                     current_statement_begin__ = 67;
522:                     stan::model::assign(U_transpose,
523:                             stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(j), stan::model::nil_index_list())),
524:                             Uj_tmp,
525:                             "assigning variable U_transpose");
526:                     current_statement_begin__ = 68;
527:                     stan::math::assign(index, (index + 1));
528:                 }
529:                 current_statement_begin__ = 70;
530:                 stan::math::assign(index, 1);
531:             } else {
532:
533:                 current_statement_begin__ = 72;
534:                 for (int j = (k + 1); j <= TT; ++j) {
535:
536:                     current_statement_begin__ = 73;
537:                     stan::math::assign(cos1, stan::math::cos(get_base1(theta_i2toQ, index,
"theta_i2toQ", 1)));
```

```
538:                    current_statement_begin__ = 74;
539:                    stan::math::assign(sin1, stan::math::sin(get_base1(theta_i2toQ, index,
"theta_i2toQ", 1)));
540:                    current_statement_begin__ = 75;
541:                    stan::math::assign(Ui_tmp, subtract(multiply(cos1,
stan::model::rvalue(U_transpose, stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(k), stan::model::nil_index_list())), "U_transpose")),
multiply(sin1, stan::model::rvalue(U_transpose,
stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(j), stan::model::nil_index_list())),
"U_transpose"))));
542:                    current_statement_begin__ = 76;
543:                    stan::math::assign(Uj_tmp, add(multiply(sin1,
stan::model::rvalue(U_transpose, stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(k), stan::model::nil_index_list())), "U_transpose")),
multiply(cos1, stan::model::rvalue(U_transpose,
stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(j), stan::model::nil_index_list())),
"U_transpose"))));
544:                    current_statement_begin__ = 77;
545:                    stan::model::assign(U_transpose,
546:                            stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(k), stan::model::nil_index_list())),
547:                            Ui_tmp,
548:                            "assigning variable U_transpose");
549:                    current_statement_begin__ = 78;
550:                    stan::model::assign(U_transpose,
551:                            stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(j), stan::model::nil_index_list())),
552:                            Uj_tmp,
553:                            "assigning variable U_transpose");
554:                    current_statement_begin__ = 79;
555:                    stan::math::assign(index, (index + 1));
556:                }
557:            }
558:            current_statement_begin__ = 83;
559:            stan::model::assign(sum_gamma,
560:                    stan::model::cons_list(stan::model::index_uni(k),
stan::model::nil_index_list()),
561:                    (1 + sum(gamma_w_seg)),
562:                    "assigning variable sum_gamma");
563:            current_statement_begin__ = 84;
564:            stan::model::assign(w,
565:
stan::model::cons_list(stan::model::index_min_max(get_base1(start_pos_GMMpara, k,
"start_pos_GMMpara", 1), (get_base1(start_pos_GMMpara, (k + 1), "start_pos_GMMpara", 1) -
1)), stan::model::nil_index_list()),
566:                    elt_divide(append_row(1.0, gamma_w_seg), get_base1(sum_gamma, k,
"sum_gamma", 1)),
```

```
567:                         "assigning variable w");
568:               }
569:             }
570:             }
571:
572:             // validate transformed parameters
573:             const char* function__ = "validate transformed params";
574:             (void) function__;  // dummy to suppress unused var warning
575:
576:             current_statement_begin__ = 39;
577:             size_t U_transpose_j_1_max__ = TT;
578:             size_t U_transpose_j_2_max__ = TT;
579:             for (size_t j_1__ = 0; j_1__ < U_transpose_j_1_max__; ++j_1__) {
580:               for (size_t j_2__ = 0; j_2__ < U_transpose_j_2_max__; ++j_2__) {
581:                 if (stan::math::is_uninitialized(U_transpose(j_1__, j_2__))) {
582:                   std::stringstream msg__;
583:                   msg__ << "Undefined transformed parameter: U_transpose" << "(" << j_1__
<< ", " << j_2__ << ")";
584:                   stan::lang::rethrow_located(std::runtime_error(std::string("Error initializing
variable U_transpose: ") + msg__.str()), current_statement_begin__, prog_reader__());
585:                 }
586:               }
587:             }
588:             current_statement_begin__ = 40;
589:             size_t w_j_1_max__ = sum_C;
590:             for (size_t j_1__ = 0; j_1__ < w_j_1_max__; ++j_1__) {
591:               if (stan::math::is_uninitialized(w(j_1__))) {
592:                 std::stringstream msg__;
593:                 msg__ << "Undefined transformed parameter: w" << "(" << j_1__ << ")";
594:                 stan::lang::rethrow_located(std::runtime_error(std::string("Error initializing
variable w: ") + msg__.str()), current_statement_begin__, prog_reader__());
595:               }
596:             }
597:             check_greater_or_equal(function__, "w", w, 0);
598:
599:             current_statement_begin__ = 44;
600:             size_t sum_gamma_j_1_max__ = Q;
601:             for (size_t j_1__ = 0; j_1__ < sum_gamma_j_1_max__; ++j_1__) {
602:               if (stan::math::is_uninitialized(sum_gamma(j_1__))) {
603:                 std::stringstream msg__;
604:                 msg__ << "Undefined transformed parameter: sum_gamma" << "(" << j_1__
<< ")";
605:                 stan::lang::rethrow_located(std::runtime_error(std::string("Error initializing
variable sum_gamma: ") + msg__.str()), current_statement_begin__, prog_reader__());
606:               }
607:             }
608:
609:             // model body
610:             {
```

```
611:            current_statement_begin__ = 89;
612:            validate_non_negative_index("S", "V", V);
613:            validate_non_negative_index("S", "Q", Q);
614:            Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, Eigen::Dynamic> S(V, Q);
615:            stan::math::initialize(S, DUMMY_VAR__);
616:            stan::math::fill(S, DUMMY_VAR__);
617:            stan::math::assign(S,multiply(X, stan::model::rvalue(U_transpose,
stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_min_max(1, Q), stan::model::nil_index_list())),
"U_transpose")));
618:
619:
620:            current_statement_begin__ = 95;
621:            for (int k = 1; k <= Q; ++k) {
622:              {
623:                current_statement_begin__ = 96;
624:                validate_non_negative_index("w_seg", "get_base1(C, k, \"C\", 1)", get_base1(C,
k, "C", 1));
625:                Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> w_seg(get_base1(C, k, "C",
1));
626:                stan::math::initialize(w_seg, DUMMY_VAR__);
627:                stan::math::fill(w_seg, DUMMY_VAR__);
628:                stan::math::assign(w_seg,segment(w, get_base1(start_pos_GMMpara, k,
"start_pos_GMMpara", 1), get_base1(C, k, "C", 1)));
629:
630:                current_statement_begin__ = 97;
631:                validate_non_negative_index("ln_w_seg", "get_base1(C, k, \"C\", 1)",
get_base1(C, k, "C", 1));
632:                Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> ln_w_seg(get_base1(C, k,
"C", 1));
633:                stan::math::initialize(ln_w_seg, DUMMY_VAR__);
634:                stan::math::fill(ln_w_seg, DUMMY_VAR__);
635:                stan::math::assign(ln_w_seg,stan::math::log(w_seg));
636:
637:                current_statement_begin__ = 98;
638:                validate_non_negative_index("mu_free_seg", "get_base1(C, k, \"C\", 1)",
get_base1(C, k, "C", 1));
639:                Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> mu_free_seg(get_base1(C,
k, "C", 1));
640:                stan::math::initialize(mu_free_seg, DUMMY_VAR__);
641:                stan::math::fill(mu_free_seg, DUMMY_VAR__);
642:                stan::math::assign(mu_free_seg,segment(mu_free,
get_base1(start_pos_GMMpara, k, "start_pos_GMMpara", 1), get_base1(C, k, "C", 1)));
643:
644:                current_statement_begin__ = 99;
645:                validate_non_negative_index("log_sigma_free_sq_seg", "get_base1(C, k, \"C\",
1)", get_base1(C, k, "C", 1));
646:                Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1>
log_sigma_free_sq_seg(get_base1(C, k, "C", 1));
```

```
647:             stan::math::initialize(log_sigma_free_sq_seg, DUMMY_VAR__);
648:             stan::math::fill(log_sigma_free_sq_seg, DUMMY_VAR__);
649:             stan::math::assign(log_sigma_free_sq_seg,segment(log_sigma_free_sq,
get_base1(start_pos_GMMpara, k, "start_pos_GMMpara", 1), get_base1(C, k, "C", 1)));
650:
651:             current_statement_begin__ = 100;
652:             validate_non_negative_index("sigma_free_seg", "get_base1(C, k, \"C\", 1)",
get_base1(C, k, "C", 1));
653:             Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1>
sigma_free_seg(get_base1(C, k, "C", 1));
654:             stan::math::initialize(sigma_free_seg, DUMMY_VAR__);
655:             stan::math::fill(sigma_free_seg, DUMMY_VAR__);
656:
stan::math::assign(sigma_free_seg,stan::math::exp(divide(log_sigma_free_sq_seg, 2)));
657:
658:             current_statement_begin__ = 102;
659:             validate_non_negative_index("loglik_v", "V", V);
660:             Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> loglik_v(V);
661:             stan::math::initialize(loglik_v, DUMMY_VAR__);
662:             stan::math::fill(loglik_v, DUMMY_VAR__);
663:
664:
665:             current_statement_begin__ = 108;
666:             lp_accum__.add(dirichlet_log<propto__>(w_seg, rep_vector(delta_w,
get_base1(C, k, "C", 1))));
667:             current_statement_begin__ = 113;
668:             lp_accum__.add((-(get_base1(C, k, "C", 1)) *
stan::math::log(get_base1(sum_gamma, k, "sum_gamma", 1))));
669:             current_statement_begin__ = 120;
670:             lp_accum__.add(normal_log<propto__>(mu_free_seg, prior_mean_mu_free,
prior_sd_mu_free));
671:             current_statement_begin__ = 127;
672:             lp_accum__.add(normal_log<propto__>(log_sigma_free_sq_seg,
prior_mean_log_sigma_free_sq, prior_sd_log_sigma_free_sq));
673:             current_statement_begin__ = 134;
674:             for (int v = 1; v <= V; ++v) {
675:               {
676:               current_statement_begin__ = 135;
677:               validate_non_negative_index("lps", "get_base1(C, k, \"C\", 1)", get_base1(C,
k, "C", 1));
678:               Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> lps(get_base1(C, k, "C",
1));
679:               stan::math::initialize(lps, DUMMY_VAR__);
680:               stan::math::fill(lps, DUMMY_VAR__);
681:               stan::math::assign(lps,ln_w_seg);
682:
683:
684:               current_statement_begin__ = 136;
685:               for (int cc = 1; cc <= get_base1(C, k, "C", 1); ++cc) {
```

```
686:
687:                     current_statement_begin__ = 137;
688:                     stan::model::assign(lps,
689:                             stan::model::cons_list(stan::model::index_uni(cc),
stan::model::nil_index_list()),
690:                             (stan::model::rvalue(lps,
stan::model::cons_list(stan::model::index_uni(cc), stan::model::nil_index_list()), "lps") +
normal_log(get_base1(S, v, k, "S", 1), get_base1(mu_free_seg, cc, "mu_free_seg", 1),
get_base1(sigma_free_seg, cc, "sigma_free_seg", 1))),
691:                             "assigning variable lps");
692:                 }
693:                 current_statement_begin__ = 139;
694:                 stan::model::assign(loglik_v,
695:                         stan::model::cons_list(stan::model::index_uni(v),
stan::model::nil_index_list()),
696:                         log_sum_exp(lps),
697:                         "assigning variable loglik_v");
698:                 }
699:             }
700:             current_statement_begin__ = 151;
701:             lp_accum__.add(sum(loglik_v));
702:             }
703:         }
704:         }
705:
706:     } catch (const std::exception& e) {
707:         stan::lang::rethrow_located(e, current_statement_begin__, prog_reader__());
708:         // Next line prevents compiler griping about no return
709:         throw std::runtime_error("*** IF YOU SEE THIS, PLEASE REPORT A BUG ***");
710:     }
711:
712:     lp_accum__.add(lp__);
713:     return lp_accum__.sum();
714:
715: } // log_prob()
716:
717: template <bool propto, bool jacobian, typename T_>
718: T_ log_prob(Eigen::Matrix<T_,Eigen::Dynamic,1>& params_r,
719:             std::ostream* pstream = 0) const {
720:   std::vector<T_> vec_params_r;
721:   vec_params_r.reserve(params_r.size());
722:   for (int i = 0; i < params_r.size(); ++i)
723:     vec_params_r.push_back(params_r(i));
724:   std::vector<int> vec_params_i;
725:   return log_prob<propto,jacobian,T_>(vec_params_r, vec_params_i, pstream);
726: }
727:
728:
729: void get_param_names(std::vector<std::string>& names__) const {
```

```
730:       names__.resize(0);
731:       names__.push_back("theta_i1");
732:       names__.push_back("theta_i2toQ");
733:       names__.push_back("gamma_w");
734:       names__.push_back("mu_free");
735:       names__.push_back("log_sigma_free_sq");
736:       names__.push_back("U_transpose");
737:       names__.push_back("w");
738:       names__.push_back("sum_gamma");
739:   }
740:
741:
742:   void get_dims(std::vector<std::vector<size_t> >& dimss__) const {
743:       dimss__.resize(0);
744:       std::vector<size_t> dims__;
745:       dims__.resize(0);
746:       dims__.push_back((TT - 1));
747:       dimss__.push_back(dims__);
748:       dims__.resize(0);
749:       dims__.push_back(dim_theta_i2toQ);
750:       dimss__.push_back(dims__);
751:       dims__.resize(0);
752:       dims__.push_back((sum_C - Q));
753:       dimss__.push_back(dims__);
754:       dims__.resize(0);
755:       dims__.push_back(sum_C);
756:       dimss__.push_back(dims__);
757:       dims__.resize(0);
758:       dims__.push_back(sum_C);
759:       dimss__.push_back(dims__);
760:       dims__.resize(0);
761:       dims__.push_back(TT);
762:       dims__.push_back(TT);
763:       dimss__.push_back(dims__);
764:       dims__.resize(0);
765:       dims__.push_back(sum_C);
766:       dimss__.push_back(dims__);
767:       dims__.resize(0);
768:       dims__.push_back(Q);
769:       dimss__.push_back(dims__);
770:   }
771:
772:   template <typename RNG>
773:   void write_array(RNG& base_rng__,
774:                    std::vector<double>& params_r__,
775:                    std::vector<int>& params_i__,
776:                    std::vector<double>& vars__,
777:                    bool include_tparams__ = true,
778:                    bool include_gqs__ = true,
```

```
779:                     std::ostream* pstream__ = 0) const {
780:       typedef double local_scalar_t__;
781:
782:       vars__.resize(0);
783:       stan::io::reader<local_scalar_t__> in__(params_r__, params_i__);
784:       static const char* function__ =
"model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::write_array";
785:       (void) function__;  // dummy to suppress unused var warning
786:
787:       // read-transform, write parameters
788:       Eigen::Matrix<double, Eigen::Dynamic, 1> theta_i1 = in__.vector_lub_constrain(-
(stan::math::pi()), stan::math::pi(), (TT - 1));
789:       size_t theta_i1_j_1_max__ = (TT - 1);
790:       for (size_t j_1__ = 0; j_1__ < theta_i1_j_1_max__; ++j_1__) {
791:         vars__.push_back(theta_i1(j_1__));
792:       }
793:
794:       Eigen::Matrix<double, Eigen::Dynamic, 1> theta_i2toQ = in__.vector_lub_constrain(0,
stan::math::pi(), dim_theta_i2toQ);
795:       size_t theta_i2toQ_j_1_max__ = dim_theta_i2toQ;
796:       for (size_t j_1__ = 0; j_1__ < theta_i2toQ_j_1_max__; ++j_1__) {
797:         vars__.push_back(theta_i2toQ(j_1__));
798:       }
799:
800:       Eigen::Matrix<double, Eigen::Dynamic, 1> gamma_w = in__.vector_lb_constrain(0,
(sum_C - Q));
801:       size_t gamma_w_j_1_max__ = (sum_C - Q);
802:       for (size_t j_1__ = 0; j_1__ < gamma_w_j_1_max__; ++j_1__) {
803:         vars__.push_back(gamma_w(j_1__));
804:       }
805:
806:       Eigen::Matrix<double, Eigen::Dynamic, 1> mu_free = in__.vector_constrain(sum_C);
807:       size_t mu_free_j_1_max__ = sum_C;
808:       for (size_t j_1__ = 0; j_1__ < mu_free_j_1_max__; ++j_1__) {
809:         vars__.push_back(mu_free(j_1__));
810:       }
811:
812:       Eigen::Matrix<double, Eigen::Dynamic, 1> log_sigma_free_sq =
in__.vector_constrain(sum_C);
813:       size_t log_sigma_free_sq_j_1_max__ = sum_C;
814:       for (size_t j_1__ = 0; j_1__ < log_sigma_free_sq_j_1_max__; ++j_1__) {
815:         vars__.push_back(log_sigma_free_sq(j_1__));
816:       }
817:
818:       double lp__ = 0.0;
819:       (void) lp__;  // dummy to suppress unused var warning
820:       stan::math::accumulator<double> lp_accum__;
821:
822:       local_scalar_t__ DUMMY_VAR__(std::numeric_limits<double>::quiet_NaN());
```

```
823:       (void) DUMMY_VAR__;  // suppress unused var warning
824:
825:       if (!include_tparams__ && !include_gqs__) return;
826:
827:       try {
828:         // declare and define transformed parameters
829:         current_statement_begin__ = 39;
830:         validate_non_negative_index("U_transpose", "TT", TT);
831:         validate_non_negative_index("U_transpose", "TT", TT);
832:         Eigen::Matrix<double, Eigen::Dynamic, Eigen::Dynamic> U_transpose(TT, TT);
833:         stan::math::initialize(U_transpose, DUMMY_VAR__);
834:         stan::math::fill(U_transpose, DUMMY_VAR__);
835:         stan::math::assign(U_transpose,diag_matrix(rep_vector(1, TT)));
836:
837:         current_statement_begin__ = 40;
838:         validate_non_negative_index("w", "sum_C", sum_C);
839:         Eigen::Matrix<double, Eigen::Dynamic, 1> w(sum_C);
840:         stan::math::initialize(w, DUMMY_VAR__);
841:         stan::math::fill(w, DUMMY_VAR__);
842:
843:         current_statement_begin__ = 44;
844:         validate_non_negative_index("sum_gamma", "Q", Q);
845:         Eigen::Matrix<double, Eigen::Dynamic, 1> sum_gamma(Q);
846:         stan::math::initialize(sum_gamma, DUMMY_VAR__);
847:         stan::math::fill(sum_gamma, DUMMY_VAR__);
848:
849:         // do transformed parameters statements
850:         {
851:         current_statement_begin__ = 49;
852:         int index(0);
853:         (void) index;  // dummy to suppress unused var warning
854:         stan::math::fill(index, std::numeric_limits<int>::min());
855:         stan::math::assign(index,1);
856:
857:         current_statement_begin__ = 50;
858:         local_scalar_t__ cos1(DUMMY_VAR__);
859:         (void) cos1;  // dummy to suppress unused var warning
860:         stan::math::initialize(cos1, DUMMY_VAR__);
861:         stan::math::fill(cos1, DUMMY_VAR__);
862:
863:         current_statement_begin__ = 51;
864:         local_scalar_t__ sin1(DUMMY_VAR__);
865:         (void) sin1;  // dummy to suppress unused var warning
866:         stan::math::initialize(sin1, DUMMY_VAR__);
867:         stan::math::fill(sin1, DUMMY_VAR__);
868:
869:         current_statement_begin__ = 52;
870:         validate_non_negative_index("Ui_tmp", "TT", TT);
871:         Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> Ui_tmp(TT);
```

```
872:            stan::math::initialize(Ui_tmp, DUMMY_VAR__);
873:            stan::math::fill(Ui_tmp, DUMMY_VAR__);
874:
875:            current_statement_begin__ = 53;
876:            validate_non_negative_index("Uj_tmp", "TT", TT);
877:            Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1> Uj_tmp(TT);
878:            stan::math::initialize(Uj_tmp, DUMMY_VAR__);
879:            stan::math::fill(Uj_tmp, DUMMY_VAR__);
880:
881:
882:            current_statement_begin__ = 55;
883:            for (int k = 1; k <= Q; ++k) {
884:              {
885:                current_statement_begin__ = 56;
886:                validate_non_negative_index("gamma_w_seg", "(get_base1(C, k, \"C\", 1) - 1)",
(get_base1(C, k, "C", 1) - 1));
887:                Eigen::Matrix<local_scalar_t__, Eigen::Dynamic, 1>
gamma_w_seg((get_base1(C, k, "C", 1) - 1));
888:                stan::math::initialize(gamma_w_seg, DUMMY_VAR__);
889:                stan::math::fill(gamma_w_seg, DUMMY_VAR__);
890:                stan::math::assign(gamma_w_seg,segment(gamma_w,
get_base1(start_pos_gamma, k, "start_pos_gamma", 1), (get_base1(C, k, "C", 1) - 1)));
891:
892:
893:                current_statement_begin__ = 60;
894:                if (as_bool(logical_eq(k, 1))) {
895:
896:                  current_statement_begin__ = 61;
897:                  for (int j = (k + 1); j <= TT; ++j) {
898:
899:                    current_statement_begin__ = 62;
900:                    stan::math::assign(cos1, stan::math::cos(get_base1(theta_i1, index,
"theta_i1", 1)));
901:                    current_statement_begin__ = 63;
902:                    stan::math::assign(sin1, stan::math::sin(get_base1(theta_i1, index,
"theta_i1", 1)));
903:                    current_statement_begin__ = 64;
904:                    stan::math::assign(Ui_tmp, subtract(multiply(cos1,
stan::model::rvalue(U_transpose, stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(k), stan::model::nil_index_list())), "U_transpose")),
multiply(sin1, stan::model::rvalue(U_transpose,
stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(j), stan::model::nil_index_list())),
"U_transpose"))));
905:                    current_statement_begin__ = 65;
906:                    stan::math::assign(Uj_tmp, add(multiply(sin1,
stan::model::rvalue(U_transpose, stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(k), stan::model::nil_index_list())), "U_transpose")),
multiply(cos1, stan::model::rvalue(U_transpose,
```

```
stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(j), stan::model::nil_index_list())),
"U_transpose"))));
907:                     current_statement_begin__ = 66;
908:                     stan::model::assign(U_transpose,
909:                             stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(k), stan::model::nil_index_list())),
910:                             Ui_tmp,
911:                             "assigning variable U_transpose");
912:                     current_statement_begin__ = 67;
913:                     stan::model::assign(U_transpose,
914:                             stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(j), stan::model::nil_index_list())),
915:                             Uj_tmp,
916:                             "assigning variable U_transpose");
917:                     current_statement_begin__ = 68;
918:                     stan::math::assign(index, (index + 1));
919:                 }
920:                 current_statement_begin__ = 70;
921:                 stan::math::assign(index, 1);
922:             } else {
923:
924:                 current_statement_begin__ = 72;
925:                 for (int j = (k + 1); j <= TT; ++j) {
926:
927:                     current_statement_begin__ = 73;
928:                     stan::math::assign(cos1, stan::math::cos(get_base1(theta_i2toQ, index,
"theta_i2toQ", 1)));
929:                     current_statement_begin__ = 74;
930:                     stan::math::assign(sin1, stan::math::sin(get_base1(theta_i2toQ, index,
"theta_i2toQ", 1)));
931:                     current_statement_begin__ = 75;
932:                     stan::math::assign(Ui_tmp, subtract(multiply(cos1,
stan::model::rvalue(U_transpose, stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(k), stan::model::nil_index_list())), "U_transpose")),
multiply(sin1, stan::model::rvalue(U_transpose,
stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(j), stan::model::nil_index_list())),
"U_transpose"))));
933:                     current_statement_begin__ = 76;
934:                     stan::math::assign(Uj_tmp, add(multiply(sin1,
stan::model::rvalue(U_transpose, stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(k), stan::model::nil_index_list())), "U_transpose")),
multiply(cos1, stan::model::rvalue(U_transpose,
stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(j), stan::model::nil_index_list())),
"U_transpose"))));
935:                     current_statement_begin__ = 77;
936:                     stan::model::assign(U_transpose,
```

```
937:                                 stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(k), stan::model::nil_index_list())),
938:                                 Ui_tmp,
939:                                 "assigning variable U_transpose");
940:                         current_statement_begin__ = 78;
941:                         stan::model::assign(U_transpose,
942:                                 stan::model::cons_list(stan::model::index_omni(),
stan::model::cons_list(stan::model::index_uni(j), stan::model::nil_index_list())),
943:                                 Uj_tmp,
944:                                 "assigning variable U_transpose");
945:                         current_statement_begin__ = 79;
946:                         stan::math::assign(index, (index + 1));
947:                     }
948:                 }
949:                 current_statement_begin__ = 83;
950:                 stan::model::assign(sum_gamma,
951:                         stan::model::cons_list(stan::model::index_uni(k),
stan::model::nil_index_list()),
952:                         (1 + sum(gamma_w_seg)),
953:                         "assigning variable sum_gamma");
954:                 current_statement_begin__ = 84;
955:                 stan::model::assign(w,
956:
stan::model::cons_list(stan::model::index_min_max(get_base1(start_pos_GMMpara, k,
"start_pos_GMMpara", 1), (get_base1(start_pos_GMMpara, (k + 1), "start_pos_GMMpara", 1) -
1)), stan::model::nil_index_list()),
957:                         elt_divide(append_row(1.0, gamma_w_seg), get_base1(sum_gamma, k,
"sum_gamma", 1)),
958:                         "assigning variable w");
959:                 }
960:             }
961:             }
962:
963:             if (!include_gqs__ && !include_tparams__) return;
964:             // validate transformed parameters
965:             const char* function__ = "validate transformed params";
966:             (void) function__;  // dummy to suppress unused var warning
967:
968:             current_statement_begin__ = 40;
969:             check_greater_or_equal(function__, "w", w, 0);
970:
971:             // write transformed parameters
972:             if (include_tparams__) {
973:                 size_t U_transpose_j_2_max__ = TT;
974:                 size_t U_transpose_j_1_max__ = TT;
975:                 for (size_t j_2__ = 0; j_2__ < U_transpose_j_2_max__; ++j_2__) {
976:                     for (size_t j_1__ = 0; j_1__ < U_transpose_j_1_max__; ++j_1__) {
977:                         vars__.push_back(U_transpose(j_1__, j_2__));
978:                     }
```

```
979:             }
980:             size_t w_j_1_max__ = sum_C;
981:             for (size_t j_1__ = 0; j_1__ < w_j_1_max__; ++j_1__) {
982:                vars__.push_back(w(j_1__));
983:             }
984:             size_t sum_gamma_j_1_max__ = Q;
985:             for (size_t j_1__ = 0; j_1__ < sum_gamma_j_1_max__; ++j_1__) {
986:                vars__.push_back(sum_gamma(j_1__));
987:             }
988:          }
989:          if (!include_gqs__) return;
990:       } catch (const std::exception& e) {
991:          stan::lang::rethrow_located(e, current_statement_begin__, prog_reader__());
992:          // Next line prevents compiler griping about no return
993:          throw std::runtime_error("*** IF YOU SEE THIS, PLEASE REPORT A BUG ***");
994:       }
995:    }
996:
997:    template <typename RNG>
998:    void write_array(RNG& base_rng,
999:                     Eigen::Matrix<double,Eigen::Dynamic,1>& params_r,
1000:                     Eigen::Matrix<double,Eigen::Dynamic,1>& vars,
1001:                     bool include_tparams = true,
1002:                     bool include_gqs = true,
1003:                     std::ostream* pstream = 0) const {
1004:      std::vector<double> params_r_vec(params_r.size());
1005:      for (int i = 0; i < params_r.size(); ++i)
1006:        params_r_vec[i] = params_r(i);
1007:      std::vector<double> vars_vec;
1008:      std::vector<int> params_i_vec;
1009:      write_array(base_rng, params_r_vec, params_i_vec, vars_vec, include_tparams,
include_gqs, pstream);
1010:      vars.resize(vars_vec.size());
1011:      for (int i = 0; i < vars.size(); ++i)
1012:        vars(i) = vars_vec[i];
1013:    }
1014:
1015:    static std::string model_name() {
1016:       return "model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ";
1017:    }
1018:
1019:
1020:    void constrained_param_names(std::vector<std::string>& param_names__,
1021:                                 bool include_tparams__ = true,
1022:                                 bool include_gqs__ = true) const {
1023:       std::stringstream param_name_stream__;
1024:       size_t theta_i1_j_1_max__ = (TT - 1);
1025:       for (size_t j_1__ = 0; j_1__ < theta_i1_j_1_max__; ++j_1__) {
1026:          param_name_stream__.str(std::string());
```

```
1027:             param_name_stream__ << "theta_i1" << '.' << j_1__ + 1;
1028:             param_names__.push_back(param_name_stream__.str());
1029:         }
1030:         size_t theta_i2toQ_j_1_max__ = dim_theta_i2toQ;
1031:         for (size_t j_1__ = 0; j_1__ < theta_i2toQ_j_1_max__; ++j_1__) {
1032:             param_name_stream__.str(std::string());
1033:             param_name_stream__ << "theta_i2toQ" << '.' << j_1__ + 1;
1034:             param_names__.push_back(param_name_stream__.str());
1035:         }
1036:         size_t gamma_w_j_1_max__ = (sum_C - Q);
1037:         for (size_t j_1__ = 0; j_1__ < gamma_w_j_1_max__; ++j_1__) {
1038:             param_name_stream__.str(std::string());
1039:             param_name_stream__ << "gamma_w" << '.' << j_1__ + 1;
1040:             param_names__.push_back(param_name_stream__.str());
1041:         }
1042:         size_t mu_free_j_1_max__ = sum_C;
1043:         for (size_t j_1__ = 0; j_1__ < mu_free_j_1_max__; ++j_1__) {
1044:             param_name_stream__.str(std::string());
1045:             param_name_stream__ << "mu_free" << '.' << j_1__ + 1;
1046:             param_names__.push_back(param_name_stream__.str());
1047:         }
1048:         size_t log_sigma_free_sq_j_1_max__ = sum_C;
1049:         for (size_t j_1__ = 0; j_1__ < log_sigma_free_sq_j_1_max__; ++j_1__) {
1050:             param_name_stream__.str(std::string());
1051:             param_name_stream__ << "log_sigma_free_sq" << '.' << j_1__ + 1;
1052:             param_names__.push_back(param_name_stream__.str());
1053:         }
1054: 
1055:         if (!include_gqs__ && !include_tparams__) return;
1056: 
1057:         if (include_tparams__) {
1058:             size_t U_transpose_j_2_max__ = TT;
1059:             size_t U_transpose_j_1_max__ = TT;
1060:             for (size_t j_2__ = 0; j_2__ < U_transpose_j_2_max__; ++j_2__) {
1061:                 for (size_t j_1__ = 0; j_1__ < U_transpose_j_1_max__; ++j_1__) {
1062:                     param_name_stream__.str(std::string());
1063:                     param_name_stream__ << "U_transpose" << '.' << j_1__ + 1 << '.' << j_2__
+ 1;
1064:                     param_names__.push_back(param_name_stream__.str());
1065:                 }
1066:             }
1067:             size_t w_j_1_max__ = sum_C;
1068:             for (size_t j_1__ = 0; j_1__ < w_j_1_max__; ++j_1__) {
1069:                 param_name_stream__.str(std::string());
1070:                 param_name_stream__ << "w" << '.' << j_1__ + 1;
1071:                 param_names__.push_back(param_name_stream__.str());
1072:             }
1073:             size_t sum_gamma_j_1_max__ = Q;
1074:             for (size_t j_1__ = 0; j_1__ < sum_gamma_j_1_max__; ++j_1__) {
```

```
1075:             param_name_stream__.str(std::string());
1076:             param_name_stream__ << "sum_gamma" << '.' << j_1__ + 1;
1077:             param_names__.push_back(param_name_stream__.str());
1078:           }
1079:       }
1080:
1081:       if (!include_gqs__) return;
1082:     }
1083:
1084:
1085:     void unconstrained_param_names(std::vector<std::string>& param_names__,
1086:                                    bool include_tparams__ = true,
1087:                                    bool include_gqs__ = true) const {
1088:       std::stringstream param_name_stream__;
1089:       size_t theta_i1_j_1_max__ = (TT - 1);
1090:       for (size_t j_1__ = 0; j_1__ < theta_i1_j_1_max__; ++j_1__) {
1091:         param_name_stream__.str(std::string());
1092:         param_name_stream__ << "theta_i1" << '.' << j_1__ + 1;
1093:         param_names__.push_back(param_name_stream__.str());
1094:       }
1095:       size_t theta_i2toQ_j_1_max__ = dim_theta_i2toQ;
1096:       for (size_t j_1__ = 0; j_1__ < theta_i2toQ_j_1_max__; ++j_1__) {
1097:         param_name_stream__.str(std::string());
1098:         param_name_stream__ << "theta_i2toQ" << '.' << j_1__ + 1;
1099:         param_names__.push_back(param_name_stream__.str());
1100:       }
1101:       size_t gamma_w_j_1_max__ = (sum_C - Q);
1102:       for (size_t j_1__ = 0; j_1__ < gamma_w_j_1_max__; ++j_1__) {
1103:         param_name_stream__.str(std::string());
1104:         param_name_stream__ << "gamma_w" << '.' << j_1__ + 1;
1105:         param_names__.push_back(param_name_stream__.str());
1106:       }
1107:       size_t mu_free_j_1_max__ = sum_C;
1108:       for (size_t j_1__ = 0; j_1__ < mu_free_j_1_max__; ++j_1__) {
1109:         param_name_stream__.str(std::string());
1110:         param_name_stream__ << "mu_free" << '.' << j_1__ + 1;
1111:         param_names__.push_back(param_name_stream__.str());
1112:       }
1113:       size_t log_sigma_free_sq_j_1_max__ = sum_C;
1114:       for (size_t j_1__ = 0; j_1__ < log_sigma_free_sq_j_1_max__; ++j_1__) {
1115:         param_name_stream__.str(std::string());
1116:         param_name_stream__ << "log_sigma_free_sq" << '.' << j_1__ + 1;
1117:         param_names__.push_back(param_name_stream__.str());
1118:       }
1119:
1120:       if (!include_gqs__ && !include_tparams__) return;
1121:
1122:       if (include_tparams__) {
1123:         size_t U_transpose_j_2_max__ = TT;
```

```
1124:             size_t U_transpose_j_1_max__ = TT;
1125:             for (size_t j_2__ = 0; j_2__ < U_transpose_j_2_max__; ++j_2__) {
1126:                 for (size_t j_1__ = 0; j_1__ < U_transpose_j_1_max__; ++j_1__) {
1127:                     param_name_stream__.str(std::string());
1128:                     param_name_stream__ << "U_transpose" << '.' << j_1__ + 1 << '.' << j_2__
+ 1;
1129:                     param_names__.push_back(param_name_stream__.str());
1130:                 }
1131:             }
1132:             size_t w_j_1_max__ = sum_C;
1133:             for (size_t j_1__ = 0; j_1__ < w_j_1_max__; ++j_1__) {
1134:                 param_name_stream__.str(std::string());
1135:                 param_name_stream__ << "w" << '.' << j_1__ + 1;
1136:                 param_names__.push_back(param_name_stream__.str());
1137:             }
1138:             size_t sum_gamma_j_1_max__ = Q;
1139:             for (size_t j_1__ = 0; j_1__ < sum_gamma_j_1_max__; ++j_1__) {
1140:                 param_name_stream__.str(std::string());
1141:                 param_name_stream__ << "sum_gamma" << '.' << j_1__ + 1;
1142:                 param_names__.push_back(param_name_stream__.str());
1143:             }
1144:         }
1145:
1146:         if (!include_gqs__) return;
1147:     }
1148:
1149: }; // model
1150:
1151: }  // namespace
1152:
1153: typedef
model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model5baf7680f1_BLC
A_unconstrained_Givens_fixC_noZ stan_model;
1154:
1155: #include <rstan/rstaninc.hpp>
1156: /**
1157:  * Define Rcpp Module to expose stan_fit's functions to R.
1158:  */
1159:
RCPP_MODULE(stan_fit4model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_mod){
1160:
Rcpp::class_<rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_names
pace::model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
1161:                 boost::random::ecuyer1988>
>("stan_fit4model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ")
1162:     // .constructor<Rcpp::List>()
1163:     .constructor<SEXP, SEXP, SEXP>()
1164:     // .constructor<SEXP, SEXP>()
1165:     .method("call_sampler",
```

1166:
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::call_sampler)
1167:   .method("param_names",
1168:
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::param_names)
1169:   .method("param_names_oi",
1170:
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::param_names_oi)
1171:   .method("param_fnames_oi",
1172:
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::param_fnames_oi)
1173:   .method("param_dims",
1174:
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::param_dims)
1175:   .method("param_dims_oi",
1176:
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::param_dims_oi)
1177:   .method("update_param_oi",
1178:
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::update_param_oi)
1179:   .method("param_oi_tidx",
1180:
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::param_oi_tidx)
1181:   .method("grad_log_prob",
1182:
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::grad_log_prob)
1183:   .method("log_prob",
1184:
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ, boost::random::ecuyer1988>::log_prob)
1185:   .method("unconstrain_pars",

```
1186:
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::unconstrain_pars)
1187:     .method("constrain_pars",
1188:
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::constrain_pars)
1189:     .method("num_pars_unconstrained",
1190:
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::num_pars_unconstrained)
1191:     .method("unconstrained_param_names",
1192:
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::unconstrained_param_names)
1193:     .method("constrained_param_names",
1194:
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::constrained_param_names)
1195:     .method("standalone_gqs",
1196:
&rstan::stan_fit<model5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ_namespace::model
5baf7680f1_BLCA_unconstrained_Givens_fixC_noZ,
boost::random::ecuyer1988>::standalone_gqs)
1197:   ;
1198: }
1199:
1200: // declarations
1201: extern "C" {
1202: SEXP file5ba7775a38f( ) ;
1203: }
1204:
1205: // definition
1206:
1207: SEXP file5ba7775a38f(  ){
1208:  return Rcpp::wrap("BLCA_unconstrained_Givens_fixC_noZ");
1209: }
1210:
1211:
```

Error in compileCode(f, code, language = language, verbose = verbose) :
  Compilation ERROR, function(s)/method(s) not created! In file included from /usr/include/c++/4.8.2/type_traits:35:0,

from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/scal/meta/is_constant.hpp:4,
from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/arr/meta/is_constant_struct.hpp:4,
from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat/meta/is_constant_struct.hpp:4,
from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/prim/mat.hpp:15,
from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math/rev/mat.hpp:12,
from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/stan/math.hpp:4,
from /home/changy/R/x86_64-redhat-linux-gnu-library/3.6/StanHeaders/include/src/stan