

Speechresearch/gmisclib

API Documentation

October 27, 2010

Contents

Contents	1
1 Package gmisclib	24
1.1 Modules	24
1.2 Variables	28
2 Module gmisclib.HList	29
2.1 Functions	29
3 Module gmisclib.HTK_HMM_io	30
3.1 Functions	30
3.2 Variables	30
3.3 Class BadFormatError	31
3.3.1 Methods	31
3.3.2 Properties	31
3.4 Class vec	32
3.4.1 Methods	32
3.4.2 Properties	32
3.4.3 Class Variables	32
3.5 Class mixture	33
3.5.1 Methods	33
3.5.2 Properties	34
3.5.3 Class Variables	34
3.6 Class ref	34
3.6.1 Methods	34
3.6.2 Properties	34
3.7 Class state	35
3.7.1 Methods	35
3.7.2 Properties	35
3.8 Class hmm	35
3.8.1 Methods	36
3.8.2 Properties	36
3.9 Class hmmset	36
3.9.1 Methods	36
3.9.2 Properties	37
3.10 Class tokstream	37
3.10.1 Methods	37

3.10.2	Properties	38
3.10.3	Class Variables	38
3.11	Class NoPhonemeToFormat	39
3.11.1	Methods	39
3.11.2	Properties	39
4	Module gmisclib.HTK_MLF_io	40
4.1	Functions	40
4.2	Variables	42
4.3	Class ReferencedFileNotFound	43
4.3.1	Methods	43
4.3.2	Properties	43
4.4	Class writer	44
4.4.1	Methods	44
4.4.2	Properties	44
5	Module gmisclib.MFCCFile	45
5.1	Class MFCCFile	45
5.1.1	Methods	45
6	Module gmisclib.MLF_file	46
6.1	Variables	46
6.2	Class BadFormatError	46
6.2.1	Methods	46
6.2.2	Properties	46
6.3	Class NotInMLFFile	47
6.3.1	Methods	47
6.3.2	Properties	47
6.4	Class block_MLF_file	48
6.4.1	Methods	48
6.4.2	Properties	48
6.4.3	Class Variables	49
7	Module gmisclib.Num	50
7.1	Variables	50
8	Module gmisclib.Numeric_gpk	55
8.1	Functions	55
8.2	Variables	59
8.3	Class EdgesTooWide	60
8.3.1	Methods	60
8.3.2	Properties	60
9	Module gmisclib.accent_spec	61
9.1	Functions	61
9.2	Variables	62
9.3	Class BadMatchError	62
9.3.1	Methods	62
9.3.2	Properties	63
10	Module gmisclib.array_window	64
10.1	Functions	64
10.2	Variables	64

10.3 Class array_window	64
10.3.1 Methods	64
10.3.2 Class Variables	65
11 Module gmisclib.avio	66
11.1 Functions	66
11.2 Variables	67
11.3 Class BadFormatError	67
11.3.1 Methods	67
11.3.2 Properties	68
11.4 Class writer	68
11.4.1 Methods	68
11.4.2 Properties	69
11.4.3 Class Variables	69
12 Module gmisclib.bark_scale	70
12.1 Functions	70
12.2 Variables	70
13 Module gmisclib.beamsearch	71
13.1 Functions	71
13.2 Variables	71
14 Module gmisclib.blue_data_attributes	72
14.1 Functions	72
14.2 Variables	72
14.3 Class blue_attributes	72
14.3.1 Methods	72
14.3.2 Properties	73
15 Module gmisclib.blue_data_selector	74
15.1 Functions	74
15.2 Variables	74
15.3 Class bluedata	74
15.3.1 Methods	75
15.3.2 Properties	76
16 Module gmisclib.cache	78
16.1 Functions	78
16.2 Variables	78
17 Module gmisclib.chunkio	79
17.1 Functions	79
17.2 Variables	79
17.3 Class BadFileFormat	79
17.3.1 Methods	80
17.3.2 Properties	80
17.4 Class chunk	80
17.4.1 Methods	80
17.4.2 Class Variables	81
17.5 Class datachunk	82
17.5.1 Methods	82
17.5.2 Class Variables	83

17.6	Class stringchunk	83
17.6.1	Methods	83
17.6.2	Class Variables	85
17.7	Class chunk_w	85
17.7.1	Methods	85
17.8	Class datachunk_w	86
17.8.1	Methods	86
17.8.2	Class Variables	87
17.9	Class chunkstring_w	88
17.9.1	Methods	88
17.9.2	Class Variables	89
18	Module gmisclib.convex_hull2d	90
18.1	Functions	90
18.2	Variables	90
18.3	Class DuplicatePoints	91
18.3.1	Methods	91
18.3.2	Properties	91
19	Module gmisclib.dict_vector	92
19.1	Functions	92
19.2	Variables	92
19.3	Class dict_vector	92
19.3.1	Methods	92
19.3.2	Properties	94
19.3.3	Class Variables	95
20	Module gmisclib.dictops	96
20.1	Functions	96
20.2	Variables	98
20.3	Class BadFileFormatError	98
20.3.1	Methods	99
20.3.2	Properties	99
20.4	Class dict_of_lists	99
20.4.1	Methods	99
20.4.2	Properties	100
20.4.3	Class Variables	100
20.5	Class dict_of_sets	101
20.5.1	Methods	101
20.5.2	Properties	101
20.5.3	Class Variables	102
20.6	Class dict_of_accums	102
20.6.1	Methods	102
20.6.2	Properties	103
20.6.3	Class Variables	103
20.7	Class dict_of_averages	103
20.7.1	Methods	103
20.7.2	Properties	104
20.7.3	Class Variables	104
20.8	Class dict_of_maxes	104
20.8.1	Methods	104
20.8.2	Properties	105

20.8.3 Class Variables	105
20.9 Class dict_of_X	105
20.9.1 Methods	106
20.9.2 Properties	106
20.9.3 Class Variables	106
20.10 Class list_of_dicts	107
20.10.1 Methods	107
20.10.2 Properties	107
20.10.3 Class Variables	107
21 Module gmisclib.die	108
21.1 Functions	108
21.2 Variables	109
21.3 Class counter	110
21.3.1 Methods	110
21.3.2 Class Variables	110
22 Module gmisclib.die2	111
22.1 Functions	111
22.2 Variables	111
23 Module gmisclib.do_exec	113
23.1 Functions	113
23.2 Variables	113
24 Module gmisclib.do_pgplot	114
24.1 Variables	114
24.2 Class plot	114
24.2.1 Methods	114
25 Module gmisclib.ds9_region	115
25.1 Variables	115
25.2 Class writer	115
25.2.1 Methods	115
26 Module gmisclib.dyn_prog	116
26.1 Functions	116
26.2 Variables	116
27 Module gmisclib.edit_distance	117
27.1 Functions	117
27.2 Variables	119
27.3 Class text_cost	119
27.3.1 Methods	120
27.3.2 Properties	121
28 Module gmisclib.entropy	122
28.1 Functions	122
28.2 Variables	122
29 Module gmisclib.erb_scale	123
29.1 Functions	123
29.2 Variables	123

30 Module gmisclib.evolution	124
30.1 Variables	124
31 Module gmisclib.fake_file	125
31.1 Functions	125
31.2 Variables	125
31.3 Class file	125
31.3.1 Methods	125
32 Module gmisclib.fiat_merge	126
32.1 Functions	126
32.2 Variables	126
33 Module gmisclib.fiatio	127
33.1 Functions	128
33.2 Variables	133
33.3 Class FiatioWarning	133
33.3.1 Methods	133
33.3.2 Properties	134
33.4 Class writer	134
33.4.1 Methods	134
33.4.2 Properties	136
33.5 Class merged_writer	136
33.5.1 Methods	137
33.5.2 Properties	138
33.6 Class ConflictingColumnSpecification	139
33.6.1 Methods	139
33.6.2 Properties	139
34 Module gmisclib.find_home	140
34.1 Functions	140
34.2 Variables	140
35 Module gmisclib.find_ngram	141
35.1 Functions	141
35.2 Variables	142
36 Module gmisclib.findleak	143
36.1 Functions	143
36.2 Variables	143
37 Module gmisclib.ftest	144
37.1 Functions	144
37.2 Variables	145
38 Module gmisclib.fuzzygraph	146
38.1 Functions	146
38.2 Variables	146
39 Module gmisclib.g2_select	147
39.1 Functions	147
39.2 Variables	148
39.3 Class selector_c	148

39.3.1	Methods	148
39.3.2	Properties	149
40	Module gmisclib.g_closure	150
40.1	Variables	150
40.2	Class NotYet	150
40.2.1	Methods	150
40.3	Class ArgUnspecifiedError	150
40.3.1	Methods	151
40.3.2	Properties	151
40.4	Class Closure	151
40.4.1	Methods	151
41	Module gmisclib.g_datetime	153
41.1	Functions	153
41.2	Variables	153
42	Module gmisclib.g_ds9	154
42.1	Functions	154
42.2	Variables	155
42.3	Class UnknownProperty	155
42.3.1	Methods	156
42.3.2	Properties	156
42.4	Class execwait	156
42.4.1	Methods	156
42.4.2	Properties	157
42.5	Class ds9_io	157
42.5.1	Methods	157
42.5.2	Properties	158
42.6	Class BadTagError	158
42.6.1	Methods	158
42.6.2	Properties	159
42.7	Class Region	159
42.7.1	Methods	159
42.7.2	Properties	160
42.8	Class line	160
42.8.1	Methods	160
42.8.2	Properties	161
42.9	Class circle	161
42.9.1	Methods	161
42.9.2	Properties	162
42.10	Class point	162
42.10.1	Methods	163
42.10.2	Properties	163
42.11	Class text	164
42.11.1	Methods	164
42.11.2	Properties	164
42.12	Class ds9	165
42.12.1	Methods	165
42.12.2	Properties	167
42.12.3	Class Variables	167
42.13	Class ds9_file	168

42.13.1 Methods	168
42.13.2 Properties	168
43 Module gmisclib.g_encode	169
43.1 Functions	169
43.2 Variables	169
43.3 Class BadFormatError	169
43.3.1 Methods	169
43.3.2 Properties	170
43.4 Class encoder	170
43.4.1 Methods	170
44 Module gmisclib.g_entropy	171
44.1 Functions	171
44.2 Variables	171
45 Module gmisclib.g_exec	172
45.1 Functions	173
45.2 Variables	176
45.3 Class ExecError	176
45.3.1 Methods	177
45.3.2 Properties	177
46 Module gmisclib.g_implements	178
46.1 Functions	178
46.2 Variables	180
46.3 Class GLError	180
46.3.1 Methods	180
46.3.2 Properties	181
47 Module gmisclib.g_keyed_accum	182
47.1 Functions	182
47.2 Variables	182
47.3 Class keyed_avg	182
47.3.1 Methods	182
48 Module gmisclib.g_lin_fit	183
48.1 Functions	183
48.2 Variables	184
49 Module gmisclib.g_localfit	185
49.1 Functions	185
49.2 Variables	189
50 Module gmisclib.g_lpc	190
50.1 Functions	190
51 Module gmisclib.g_pipe	191
51.1 Functions	191
51.2 Variables	191
51.3 Class pfd	191
51.3.1 Methods	191
51.3.2 Properties	192

51.3.3 Class Variables	192
52 Module gmisclib.g_pipe_old	193
52.1 Functions	193
52.2 Variables	193
52.3 Class pfd	193
52.3.1 Methods	193
52.3.2 Properties	194
52.3.3 Class Variables	194
53 Module gmisclib.g_place_label	195
53.1 Functions	195
53.2 Variables	195
53.3 Class boxc	195
53.3.1 Methods	195
53.3.2 Properties	196
53.4 Class text_template	196
53.4.1 Methods	196
53.4.2 Properties	197
53.5 Class viewport	197
53.5.1 Methods	197
53.5.2 Properties	198
54 Module gmisclib.g_pylab	199
55 Module gmisclib.g_selector	200
55.1 Functions	200
55.2 Variables	200
55.3 Class selector	200
55.3.1 Methods	200
55.4 Class true	201
55.4.1 Methods	201
55.5 Class false	201
55.5.1 Methods	202
55.6 Class selector_op	202
55.6.1 Methods	202
55.7 Class one_selector	203
55.7.1 Methods	203
56 Module gmisclib.gpk_getopt	205
56.1 Functions	205
56.2 Variables	205
57 Module gmisclib.gpk_hdr	206
57.1 Functions	206
57.2 Variables	206
57.3 Class hdr	206
57.3.1 Methods	206
58 Module gmisclib.gpk_lapack	207
58.1 Functions	207
58.2 Class NoSolutionError	207
58.2.1 Methods	207

58.2.2 Properties	208
59 Module gmisclib.gpk_lsq	209
59.1 Functions	209
59.2 Variables	209
59.3 Class lls_base	209
59.3.1 Methods	210
59.3.2 Properties	210
59.4 Class linear_least_squares	211
59.4.1 Methods	211
59.4.2 Properties	213
59.5 Class reg_linear_least_squares	213
59.5.1 Methods	213
59.5.2 Properties	215
60 Module gmisclib.gpk_rlsq	216
60.1 Functions	216
60.2 Variables	216
60.3 Class NotEnoughData	216
60.3.1 Methods	216
60.3.2 Properties	217
60.4 Class w_linear_least_squares	217
60.4.1 Methods	218
60.4.2 Properties	219
60.5 Class robust_linear_fit	220
60.5.1 Methods	220
60.5.2 Properties	222
60.5.3 Class Variables	222
61 Module gmisclib.gpk_writer	223
61.1 Variables	223
61.2 Class writer	223
61.2.1 Methods	223
61.2.2 Properties	224
61.3 Class null_writer	224
61.3.1 Methods	224
61.3.2 Properties	225
62 Module gmisclib.gpkequations	226
62.1 Functions	226
62.2 Variables	227
62.3 Class BadParamError	227
62.3.1 Methods	227
62.3.2 Properties	227
62.4 Class NoTermsEx	228
62.4.1 Methods	228
62.4.2 Properties	228
62.5 Class eqn_c	229
62.5.1 Methods	229
63 Module gmisclib.gpkmisc	230
63.1 Functions	230

63.2	Variables	235
63.3	Class <code>threaded_readable_file</code>	235
63.3.1	Methods	235
63.3.2	Properties	236
63.3.3	Class Variables	236
63.4	Class <code>dir_lock</code>	236
63.4.1	Methods	236
63.4.2	Properties	237
63.5	Class <code>PrereqError</code>	237
63.5.1	Methods	237
63.5.2	Properties	238
64	Module <code>gmisclib.hilbert_xform</code>	239
64.1	Functions	239
64.2	Variables	239
65	Module <code>gmisclib.kl_dist</code>	240
65.1	Functions	240
65.2	Variables	241
65.3	Class <code>NoConvergenceError</code>	242
65.3.1	Methods	242
65.3.2	Properties	242
65.4	Class <code>NotMarkovError</code>	243
65.4.1	Methods	243
65.4.2	Properties	243
66	Module <code>gmisclib.lapack_dgesv</code>	244
66.1	Functions	244
67	Module <code>gmisclib.lapack_dsyeval</code>	245
67.1	Functions	245
68	Module <code>gmisclib.load_mod</code>	246
68.1	Functions	246
68.2	Variables	248
69	Module <code>gmisclib.lreg_fill</code>	249
69.1	Functions	249
69.2	Variables	249
70	Module <code>gmisclib.makemake</code>	250
70.1	Functions	250
70.2	Variables	251
70.3	Class <code>FileNotFound</code>	251
70.3.1	Methods	251
70.3.2	Properties	252
71	Module <code>gmisclib.matrix_arrange</code>	253
71.1	Functions	253
71.2	Variables	253
72	Module <code>gmisclib.matrix_arrange_entropy</code>	254
72.1	Functions	254

72.2 Variables	254
73 Module gmisclib.matrix_rearrange_labels	255
73.1 Functions	255
73.2 Variables	260
73.3 Class diagfom	260
73.3.1 Methods	260
73.3.2 Properties	261
73.4 Class diagfom2	261
73.4.1 Methods	261
73.4.2 Properties	262
73.5 Class blockfom	262
73.5.1 Methods	262
73.5.2 Properties	262
74 Module gmisclib.mcmc	263
74.1 Functions	263
74.2 Variables	264
74.3 Class problem_definition	264
74.3.1 Methods	265
74.3.2 Properties	267
74.4 Class position_base	267
74.4.1 Methods	267
74.4.2 Properties	269
74.5 Class position_repeatable	269
74.5.1 Methods	269
74.5.2 Properties	271
74.5.3 Class Variables	271
74.6 Class position_nonrepeatable	271
74.6.1 Methods	272
74.6.2 Properties	273
74.6.3 Class Variables	273
74.7 Class position_history_dependent	274
74.7.1 Methods	274
74.7.2 Properties	276
74.7.3 Class Variables	276
74.8 Class T_acceptor	276
74.8.1 Methods	276
74.8.2 Properties	277
74.9 Class stepper	277
74.9.1 Methods	277
74.9.2 Properties	278
74.10Class adjuster	279
74.10.1Methods	279
74.10.2Properties	280
74.10.3Class Variables	280
74.11Class NoBoot	280
74.11.1Methods	280
74.11.2Properties	281
74.12Class NotGoodPosition	281
74.12.1Methods	281
74.12.2Properties	282

74.13	Class hashcounter_c	282
74.13.1	Methods	282
74.13.2	Properties	282
74.13.3	Class Variables	283
74.14	Class Archive	283
74.14.1	Methods	283
74.14.2	Properties	284
74.14.3	Class Variables	284
74.15	Class ContPrmArchive	285
74.15.1	Methods	285
74.15.2	Properties	287
74.15.3	Class Variables	287
74.16	Class BootStepper	287
74.16.1	Methods	287
74.16.2	Properties	289
74.16.3	Class Variables	289
75	Module gmisclib.mcmcS2	290
75.1	Functions	290
76	Module gmisclib.mcmcSQ	291
76.1	Functions	291
76.2	Variables	292
76.3	Class CantGetPoint	292
76.3.1	Methods	292
76.3.2	Properties	293
76.4	Class DontGoHere	293
76.4.1	Methods	293
76.4.2	Properties	294
76.5	Class NegativeEigenvalue	294
76.5.1	Methods	294
76.5.2	Properties	295
76.6	Class BadlyConditioned	295
76.6.1	Methods	295
76.6.2	Properties	296
76.7	Class timer	296
76.7.1	Methods	296
76.8	Class problem_definition	296
76.8.1	Methods	297
76.8.2	Properties	297
76.9	Class position	297
76.9.1	Methods	297
76.9.2	Properties	298
76.9.3	Class Variables	298
76.10	Class Lambda.c	299
76.10.1	Methods	299
76.10.2	Class Variables	299
76.11	Class SQstepper	300
76.11.1	Methods	300
76.11.2	Properties	301
76.12	Class archiver	302
76.12.1	Methods	302

76.12.2 Class Variables	303
77 Module gmisclib.mcmc_big	304
77.1 Functions	304
77.2 Variables	304
77.3 Class BootStepper	304
77.3.1 Methods	305
77.3.2 Properties	307
77.3.3 Class Variables	307
78 Module gmisclib.mcmc_cooperate	308
78.1 Functions	308
78.2 Variables	308
78.3 Class Barrier	308
78.3.1 Methods	309
78.3.2 Properties	309
78.4 Class Oops	309
78.4.1 Methods	310
78.4.2 Properties	310
78.5 Class LateToBarrier	310
78.5.1 Methods	311
78.5.2 Properties	311
78.6 Class connection	311
78.6.1 Methods	311
78.6.2 Properties	312
78.6.3 Class Variables	313
79 Module gmisclib.mcmc_helper	314
79.1 Functions	314
79.2 Variables	314
79.3 Class TooManyLoops	314
79.3.1 Methods	314
79.3.2 Properties	315
79.4 Class warnever	315
79.4.1 Methods	315
79.4.2 Properties	316
79.5 Class logger_template	316
79.5.1 Methods	316
79.5.2 Properties	316
79.6 Class stepper	316
79.6.1 Methods	317
79.6.2 Properties	319
79.7 Class step_acceptor	320
79.7.1 Methods	321
79.7.2 Properties	322
80 Module gmisclib.mcmc_idxr	323
80.1 Functions	323
80.2 Variables	323
80.3 Class probdist_c	323
80.3.1 Methods	323
80.3.2 Properties	324

80.4	Class Weibull	324
80.4.1	Methods	324
80.4.2	Properties	325
80.5	Class Expo	326
80.5.1	Methods	326
80.5.2	Properties	327
80.6	Class Normal	327
80.6.1	Methods	327
80.6.2	Properties	328
80.7	Class Uniform	328
80.7.1	Methods	328
80.7.2	Properties	329
80.8	Class LogNormal	330
80.8.1	Methods	330
80.8.2	Properties	331
81	Module gmisclib.mcmc.logger	332
81.1	Functions	332
81.2	Variables	332
81.3	Class WildNumber	332
81.3.1	Methods	332
81.3.2	Properties	333
81.4	Class logger_c	333
81.4.1	Methods	333
81.4.2	Properties	334
81.5	Class logger_A	335
81.5.1	Methods	335
81.5.2	Properties	336
81.6	Class logger_N	336
81.6.1	Methods	336
81.6.2	Properties	337
81.7	Class lti_c	338
81.7.1	Methods	338
81.7.2	Properties	338
81.8	Class NoDataError	338
81.8.1	Methods	339
81.8.2	Properties	339
82	Module gmisclib.mcmc.m4p	340
82.1	Functions	340
82.2	Variables	340
82.3	Class stepper	341
82.3.1	Methods	341
82.3.2	Properties	344
82.3.3	Class Variables	344
83	Module gmisclib.mcmc.mpi	345
83.1	Functions	345
83.2	Variables	345
83.3	Class stepper	345
83.3.1	Methods	345
83.3.2	Properties	349

83.3.3 Class Variables	349
84 Module gmisclib.mcmc_pypar	350
84.1 Functions	350
84.2 Class stepper	350
84.2.1 Methods	350
84.2.2 Properties	353
84.2.3 Class Variables	353
85 Module gmisclib.mcmc_socket	354
85.1 Functions	354
85.2 Variables	354
85.3 Class stepper	354
85.3.1 Methods	354
85.3.2 Properties	358
86 Module gmisclib.md_scaling	359
86.1 Functions	359
86.2 Variables	359
87 Module gmisclib.multivariate_classes	360
87.1 Functions	360
87.2 Variables	360
87.3 Class QuadraticNotNormalizable	360
87.3.1 Methods	360
87.3.2 Properties	361
87.4 Class modeldesc	361
87.4.1 Methods	361
87.4.2 Class Variables	362
87.5 Class model_with_numbers	362
87.5.1 Methods	362
87.5.2 Class Variables	363
88 Module gmisclib.multivariate_mm	364
88.1 Functions	364
88.2 Variables	364
88.3 Class datum_c	364
88.3.1 Methods	364
88.4 Class multi_mu	364
88.4.1 Methods	365
88.4.2 Class Variables	365
88.5 Class multi_mu_with_numbers	366
88.5.1 Methods	366
88.5.2 Class Variables	367
88.6 Class multi_mu_diag	367
88.6.1 Methods	367
88.6.2 Class Variables	368
88.7 Class multi_mu_diag_with_numbers	368
88.7.1 Methods	368
88.7.2 Class Variables	369
89 Module gmisclib.multivariate_q	370

89.1	Variables	370
89.2	Class quadratic	370
89.2.1	Methods	370
89.2.2	Class Variables	371
89.3	Class quadratic_with_numbers	371
89.3.1	Methods	371
89.3.2	Class Variables	372
89.4	Class diag_quadratic	372
89.4.1	Methods	372
89.4.2	Class Variables	373
89.5	Class diag_quadratic_with_numbers	374
89.5.1	Methods	374
89.5.2	Class Variables	375
90	Module gmisclib.multivariate_normal	376
90.1	Functions	376
90.2	Variables	376
90.3	Class multivariate_normal	376
90.3.1	Methods	376
90.3.2	Class Variables	376
91	Module gmisclib.named_block_file	377
91.1	Functions	377
91.2	Variables	377
92	Module gmisclib.nbest	378
92.1	Functions	378
92.2	Variables	378
92.3	Class node	378
92.3.1	Methods	378
92.3.2	Class Variables	378
93	Module gmisclib.nice_hash	380
93.1	Functions	380
93.2	Variables	380
93.3	Class DontHashThis	380
93.3.1	Methods	380
93.3.2	Properties	381
93.4	Class NotInHash	381
93.4.1	Methods	381
93.4.2	Properties	382
93.5	Class simple	382
93.5.1	Methods	382
93.5.2	Properties	383
93.5.3	Class Variables	383
93.6	Class hash	384
93.6.1	Methods	384
93.6.2	Properties	385
93.6.3	Class Variables	386
93.7	Class hash	386
93.7.1	Methods	386
93.7.2	Properties	387

93.7.3 Class Variables	388
94 Module gmisclib.nicknames	389
94.1 Functions	389
94.2 Variables	389
95 Module gmisclib.nmf	390
95.1 Functions	390
95.2 Variables	390
96 Module gmisclib.opt	391
96.1 Functions	391
96.2 Variables	394
96.3 Class OptError	394
96.3.1 Methods	394
96.3.2 Properties	395
96.4 Class NoDownhill	395
96.4.1 Methods	395
96.4.2 Properties	396
96.5 Class NoDerivative	396
96.5.1 Methods	396
96.5.2 Properties	397
96.6 Class BadParamError	397
96.6.1 Methods	397
96.6.2 Properties	398
96.6.3 Class Variables	398
96.7 Class BadResult	398
96.7.1 Methods	398
96.7.2 Properties	399
96.7.3 Class Variables	399
96.8 Class mysem	399
96.8.1 Methods	399
96.8.2 Class Variables	399
96.9 Class semclass	400
96.9.1 Methods	400
96.9.2 Class Variables	400
96.10 Class prms	400
96.10.1 Methods	400
96.10.2 Class Variables	401
96.11 Class LockedList	401
96.11.1 Methods	401
96.12 Class opt	401
96.12.1 Methods	402
96.12.2 Class Variables	404
97 Module gmisclib.ortho_poly	405
97.1 Functions	405
97.2 Variables	405
97.3 Class ortho	405
97.3.1 Methods	405
97.3.2 Class Variables	406
97.4 Class ortho_poly	406

97.4.1	Methods	407
97.4.2	Class Variables	407
97.5	Class Legendre	408
97.5.1	Methods	408
97.5.2	Class Variables	409
97.6	Class Chebyshev	409
97.6.1	Methods	410
97.6.2	Class Variables	410
97.7	Class Chebyshev2	411
97.7.1	Methods	411
97.7.2	Class Variables	412
97.8	Class SinCos	412
97.8.1	Methods	413
97.8.2	Class Variables	413
97.9	Class SLTB	414
97.9.1	Methods	414
97.9.2	Class Variables	415
98	Module gmisclib.parse_tree_number	416
98.1	Functions	416
98.2	Variables	416
98.3	Class Array	416
98.3.1	Methods	417
98.3.2	Properties	417
98.4	Class output_mixin	418
98.4.1	Methods	418
98.4.2	Properties	418
98.5	Class abstract_number	419
98.5.1	Methods	419
98.5.2	Properties	420
98.6	Class Expression	420
98.6.1	Methods	420
98.6.2	Properties	422
98.7	Class Name	423
98.7.1	Methods	423
98.7.2	Properties	425
98.8	Class Float	425
98.8.1	Methods	425
98.8.2	Properties	427
98.9	Class operator1	427
98.9.1	Methods	428
98.9.2	Properties	429
98.9.3	Class Variables	430
98.10	Class elementof	430
98.10.1	Methods	430
98.10.2	Properties	432
98.10.3	Class Variables	432
98.11	Class operatorNg1	433
98.11.1	Methods	433
98.11.2	Properties	435
98.11.3	Class Variables	435

99	Module <code>gmisclib.permute</code>	436
99.1	Functions	436
99.2	Variables	436
100	Module <code>gmisclib.pylab_oneaxis</code>	437
100.1	Functions	437
101	Module <code>gmisclib.pylab_starplot</code>	438
101.1	Functions	438
101.2	Variables	438
101.3	Class <code>errorbar_maker</code>	438
101.3.1	Methods	439
101.3.2	Properties	439
101.4	Class <code>errorbar</code>	439
101.4.1	Methods	439
101.4.2	Properties	440
102	Module <code>gmisclib.read_dicom</code>	441
102.1	Functions	441
102.2	Variables	441
102.3	Class <code>img_with_mx</code>	441
102.3.1	Methods	441
102.3.2	Properties	442
103	Module <code>gmisclib.resample</code>	443
103.1	Functions	443
103.2	Variables	443
104	Module <code>gmisclib.robust_multivariate</code>	444
104.1	Functions	444
104.2	Variables	444
105	Module <code>gmisclib.root</code>	445
105.1	Functions	445
105.2	Variables	445
106	Module <code>gmisclib.rubber_array</code>	446
106.1	Variables	446
106.2	Class <code>ext_block</code>	446
106.2.1	Methods	446
106.2.2	Properties	446
106.3	Class <code>extensible_array</code>	447
106.3.1	Methods	447
106.3.2	Properties	447
107	Module <code>gmisclib.s_lin_fit</code>	448
107.1	Functions	448
107.2	Variables	448
107.3	Class <code>Error</code>	448
107.3.1	Methods	449
107.3.2	Properties	449
107.4	Class <code>NoDataError</code>	449
107.4.1	Methods	450

107.4.2 Properties	450
108 Module gmisclib.sbd_array	451
108.1 Functions	451
108.2 Variables	451
108.3 Class sbd	452
108.3.1 Methods	452
108.4 Class NoSolutionError	453
108.4.1 Methods	453
108.4.2 Properties	453
109 Module gmisclib.segmentfile	454
109.1 Functions	454
109.2 Variables	454
109.3 Class segment	454
109.3.1 Methods	454
109.4 Class group	455
109.4.1 Methods	455
110 Module gmisclib.sharp_energy	456
110.1 Functions	456
111 Module gmisclib.solve_sum_abs	457
111.1 Functions	457
111.2 Variables	458
112 Module gmisclib.spread_jobs	459
112.1 Functions	460
112.2 Variables	461
112.3 Class notComputed	461
112.3.1 Methods	461
112.3.2 Properties	461
112.4 Class NoResponse	462
112.4.1 Methods	462
112.4.2 Properties	462
112.5 Class RemoteException	463
112.5.1 Methods	463
112.5.2 Properties	463
112.6 Class unpickled_pseudofile	464
112.6.1 Methods	464
113 Module gmisclib.stats	465
113.1 Functions	465
113.2 Variables	466
114 Module gmisclib.synch_demod	467
114.1 Functions	467
115 Module gmisclib.threaded_io	468
115.1 Functions	468
115.2 Variables	468
115.3 Class CaughtException	469
115.3.1 Methods	469

115.3.2 Properties	469
115.4 Class threading_with_result	469
115.4.1 Methods	470
115.4.2 Properties	470
115.5 Class Thread_poolW	470
115.5.1 Methods	471
115.5.2 Properties	471
115.6 Class Thread_poolW	471
115.6.1 Methods	471
115.6.2 Properties	472
115.7 Class Thread_poolR	472
115.7.1 Methods	472
115.7.2 Properties	473
116 Module gmisclib.tsops	474
116.1 Functions	474
116.2 Class axis	476
116.2.1 Methods	476
117 Module gmisclib.wavesurfer_lab	477
117.1 Functions	477
117.2 Variables	477
118 Module gmisclib.wavio	479
118.1 Functions	479
118.2 Variables	480
118.3 Class Overflow	481
118.3.1 Methods	481
118.3.2 Properties	481
118.4 Class BadFileFormatError	482
118.4.1 Methods	482
118.4.2 Properties	482
119 Module gmisclib.weighted_percentile	483
119.1 Functions	483
119.2 Variables	484
120 Module gmisclib.xwaves_errs	485
120.1 Variables	485
120.2 Class Error	485
120.2.1 Methods	485
120.2.2 Properties	486
120.3 Class NoSuchFileError	486
120.3.1 Methods	486
120.3.2 Properties	487
120.4 Class BadFileFormatError	487
120.4.1 Methods	487
120.4.2 Properties	488
120.5 Class DataError	488
120.5.1 Methods	488
120.5.2 Properties	489
120.6 Class DataOutOfOrderError	489

120.6.1 Methods	489
120.6.2 Properties	490
121Module gmisclib.xwaves_lab	491
121.1Functions	491
121.2Variables	491
122Module gmisclib.xwaves_mark	492
122.1Functions	492
122.2Variables	492
123Module gpk_wavio	493
124Module mcmc	494
124.1Functions	496
124.2Variables	496
124.3Class def.logger	496
124.3.1 Methods	496
125Module q3html	497
125.1Functions	498
125.2Variables	499
125.3Class lineC	500
125.3.1 Methods	500
126Module run_several	501
126.1Functions	502
126.2Variables	502
127Module select_fiat_entries	504
127.1Functions	504
127.2Variables	504
127.3Class writer	505
127.3.1 Methods	505
127.3.2 Properties	506
Index	507

1 Package gmisclib

1.1 Modules

- **HList**: This reads HTK feature vectors via the HList program.
(Section 2, p. 29)
- **HTK_HMM_io**: This reads in HMM files produced by HTK.
(Section 3, p. 30)
- **HTK_MLF_io**: This reads MLF (Master Label Files) for/from the HTK speech recognition toolkit.
(Section 4, p. 40)
- **MFCCFile** (Section 5, p. 45)
- **MLF_file** (Section 6, p. 46)
- **Num**: This is a compatibility module, allowing python to work with either scipy, Numeric, or numarray.
(Section 7, p. 50)
- **Numeric_gpk** (Section 8, p. 55)
- **accent_spec**: This module provides a way of safely specifying accent positions in running text.
(Section 9, p. 61)
- **array_window**: Defines array_window class.
(Section 10, p. 64)
- **avio**: Module to parse and create lines in "a=v;" format.
(Section 11, p. 66)
- **bark_scale** (Section 12, p. 70)
- **beamsearch** (Section 13, p. 71)
- **blue_data_attributes**: This chooses samples such that the specified attributes are broadly distributed.
(Section 14, p. 72)
- **blue_data_selector**: This takes anticorrelated samples from a sequence of data.
(Section 15, p. 74)
- **cache** (Section 16, p. 78)
- **chunkio**: These are I/O routines to allow you to write stuff like arrays and dictionaries (and arrays of dictionaries) to a human-readable file.
(Section 17, p. 79)
- **convex_hull2d**: convexhull.py
(Section 18, p. 90)
- **dict_vector**: Vectors of numbers, but indexed as a dictionary.
(Section 19, p. 92)
- **dictops**: Operations on dictionaries.
(Section 20, p. 96)
- **die** (Section 21, p. 108)
- **die2** (Section 22, p. 111)
- **do_exec**: Execute a *nix command and capture the output.
(Section 23, p. 113)
- **do_pgplot** (Section 24, p. 114)
- **ds9_region** (Section 25, p. 115)
- **dyn_prog**: Viterbi search
(Section 26, p. 116)
- **edit_distance**: Levenshtein (edit) distance between two strings of symbols.
(Section 27, p. 117)
- **entropy** (Section 28, p. 122)
- **erb_scale**: ERB Perceptual frequency scale.
(Section 29, p. 123)
- **evolution** (Section 30, p. 124)

- **fake_file**: A class that implements a simple file in memory.
(Section 31, p. 125)
- **fiat_merge** (Section 32, p. 126)
- **fiatio**: Fiatio reads and writes an extension of the FIAT file format originally defined by David Wittman (UC Davis).
(Section 33, p. 127)
- **find_home**: This lets you find files with just a path name relative to where the program's executable code script sits.
(Section 34, p. 140)
- **find_ngram**: This module lets you search through label files to find particular ngrams.
(Section 35, p. 141)
- **findleak** (Section 36, p. 143)
- **ftest** (Section 37, p. 144)
- **fuzzygraph** (Section 38, p. 146)
- **g2_select** (Section 39, p. 147)
- **g_closure**: This module defines closures.
(Section 40, p. 150)
- **g_datetime** (Section 41, p. 153)
- **g_ds9**: Handles interactions with ds9 (<http://hea-www.harvard.edu/RD/ds9/>).
(Section 42, p. 154)
- **g_encode**: This module allows strings to be encoded into a reduced subset.
(Section 43, p. 169)
- **g_entropy**: This returns the entropy of a probability distribution that produced a given sample.
(Section 44, p. 171)
- **g_exec**: Run a Linux command and capture the result.
(Section 45, p. 172)
- **g_implements**: This module tells you if an object's signature matches a class.
(Section 46, p. 178)
- **g_keyed_accum** (Section 47, p. 182)
- **g_lin_fit**: Generic linear best fits.
(Section 48, p. 183)
- **g_localfit**: Fit a linear transform to a bunch of tt input/output vectors.
(Section 49, p. 185)
- **g_lpc**: Linear prediction of signals
(Section 50, p. 190)
- **g_pipe**: A multithreaded version of os.popen2().
(Section 51, p. 191)
- **g_pipe_old**: A multithreaded version of os.popen2().
(Section 52, p. 193)
- **g_place_label** (Section 53, p. 195)
- **g_pylab**: This class works in concert with bin/pylab_server.py to allow you to display simple pylab/matplotlib graphics on another machine.
(Section 54, p. 199)
- **g_selector** (Section 55, p. 200)
- **gpk_getopt** (Section 56, p. 205)
- **gpk_hdr** (Section 57, p. 206)
- **gpk_lapack** (Section 58, p. 207)
- **gpk_lsq** (Section 59, p. 209)
- **gpk_rlsq** (Section 60, p. 216)
- **gpk_writer** (Section 61, p. 223)
- **gpkequations** (Section 62, p. 226)

- **gpkmisc** (Section 63, p. 230)
- **hilbert_xform** (Section 64, p. 239)
- **kl_dist**: Suppose there is a random variable with true distribution p .
(Section 65, p. 240)
- **lapack_dgesv** (Section 66, p. 244)
- **lapack_dsyevd** (Section 67, p. 245)
- **load_mod**: This module has functions that help you dynamically import modules.
(Section 68, p. 246)
- **lreg_fill** (Section 69, p. 249)
- **makemake**: This module is designed to build makefiles.
(Section 70, p. 250)
- **matrix_arrange**: Take a covariance-like matrix, and re-order it to be close to a diagonal matrix.
(Section 71, p. 253)
- **matrix_arrange_entropy**: This rotates a matrix by multiplying with a unitary matrix so that the resulting elements are either nearly zero or relatively large.
(Section 72, p. 254)
- **matrix_rearrange_labels**: This module helps you plot confusion matrices or similar images where the axis labels do not have a natural order.
(Section 73, p. 255)
- **mcmc**: Adaptive Markov-Chain Monte-Carlo algorithm.
(Section 74, p. 263)
- **mcmcS2**: Markov-Chain Monte-Carlo algorithms.
(Section 75, p. 290)
- **mcmcSQ**: Markov-Chain Monte-Carlo algorithms.
(Section 76, p. 291)
- **mcmc_big**: An extension of mcmc that includes a new stepping algorithm.
(Section 77, p. 304)
- **mcmc_cooperate** (Section 78, p. 308)
- **mcmc_helper**: This is a helper module to make use of mcmc.py and mcmc_big.py.
(Section 79, p. 314)
- **mcmc_idxr** (Section 80, p. 323)
- **mcmc_logger** (Section 81, p. 332)
- **mcmc_m4p**: This is a helper module to make use of mcmc.py and mcmc_big.py.
(Section 82, p. 340)
- **mcmc_mpi**: This is a helper module to make use of mcmc.py and mcmc_big.py.
(Section 83, p. 345)
- **mcmc_pypar**: This is a helper module to make use of mcmc.py and mcmc_big.py.
(Section 84, p. 350)
- **mcmc_socket**: This is a helper module to make use of mcmc.py and mcmc_big.py.
(Section 85, p. 354)
- **md_scaling**: Find a set of points so that the distances between them match a desired set of distances.
(Section 86, p. 359)
- **multivariate_classes**: Support module for multivariate.py
(Section 87, p. 360)
- **multivariate_mm**: This a helper module for multivariate.py
(Section 88, p. 364)
- **multivariate_q**: This a helper module for multivariate.py
(Section 89, p. 370)
- **multivariate_normal** (Section 90, p. 376)
- **named_block_file**: Read and write files in the form [label] text text text [label2] text text ...
(Section 91, p. 377)

- **nbest**: Beam search through a graph.
(Section 92, p. 378)
- **nice_hash** (Section 93, p. 380)
- **nicknames** (Section 94, p. 389)
- **nmf**: $V=WH$, where W and H are non-negative.
(Section 95, p. 390)
- **opt**: This module is a Levenberg-Marquardt optimizer with stem-size control for numeric differentiation.
(Section 96, p. 391)
- **ortho_poly** (Section 97, p. 405)
- **parse_tree_number** (Section 98, p. 416)
- **permute**: Find different permutations of an array.
(Section 99, p. 436)
- **pylab_oneaxis**: You can control the axis tick and grid properties
(Section 100, p. 437)
- **pylab_starplot**: This makes a little 'star' of error bars, using pylab/matplotlib.
(Section 101, p. 438)
- **read_dicom** (Section 102, p. 441)
- **resample**: Resample a time series by a factor r .
(Section 103, p. 443)
- **robust_multivariate**: Does a robust estimate of covariance.
(Section 104, p. 444)
- **root**: Solve a 1-dimensional equation to find the roots.
(Section 105, p. 445)
- **rubber_array**: This acts like a dictionary, but indexed by integers and stored rather more efficiently, at least in terms of memory.
(Section 106, p. 446)
- **s_lin_fit**: Fit a plane to some data.
(Section 107, p. 448)
- **sbd_array** (Section 108, p. 451)
- **segmentfile**: This parses a '.in' file from xwaves.
(Section 109, p. 454)
- **sharp_energy** (Section 110, p. 456)
- **solve_sum_abs**: Solves various equations involving minimizing the sum of absolute values of things.
(Section 111, p. 457)
- **spread_jobs**: A module that starts a bunch of subprocesses and distributes work amongst them, then collects the results.
(Section 112, p. 459)
- **stats**: Some of these functions, specifically `f.value()`, `fprob()`, `betai()`, and `betacf()`, are taken from `stats.py` "A collection of basic statistical functions for python." by Gary Strangman.
(Section 113, p. 465)
- **synch_demod** (Section 114, p. 467)
- **threaded_io**: This is designed to let you do asynchronous I/O conveniently.
(Section 115, p. 468)
- **tsops**: Do mathematical operations on time series, where the two operands don't necessarily have the same sampling times.
(Section 116, p. 474)
- **wavesurfer_lab**: When used as a script, this reads label files produced by wavesurfer and prints the result.
(Section 117, p. 477)
- **wavio**: When run as a script: `python ~/lib/wavio.py [-g gain] -wavin|-wavout infile outfile` Reads or

writes .wav files from any format supported by gpkimclass.py .

(Section 118, p. 479)

- **weighted_percentile**: This computes order statistics on data with weights.
(Section 119, p. 483)
- **xwaves_errs**: Errors for reading label (typically speech transcription) files.
(Section 120, p. 485)
- **xwaves_lab**: Reads label files produced by ESPS xwaves.
(Section 121, p. 491)
- **xwaves_mark**: Read in .in files produced by ESPS xmark.
(Section 122, p. 492)

1.2 Variables

Name	Description
--package--	Value: None

2 Module gmisclib.HList

This reads HTK feature vectors via the HList program.

2.1 Functions

<code>read(<i>fname</i>)</code>

3 Module *gmisclib.HTK_HMM_io*

This reads in HMM files produced by HTK.

3.1 Functions

deref (<i>x</i>)

read_floats_t (<i>vtype, dim, t, tok, phoneme='???'</i>)

read_floats_b (<i>vtype, dim, t, tok, phoneme='???'</i> , <i>name=None</i>)
--

read_vec (<i>dim, n, t, name=None, vtype=None</i>)

read (<i>fdlist</i>)

HTKformat (<i>s</i>)

Format for HTK state name.

dict_format (<i>s</i>)

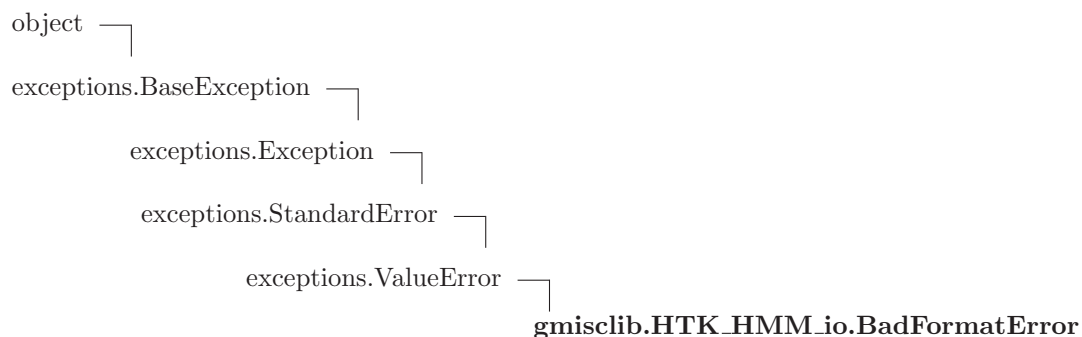
Format for center [bracket] in HTK dictionary.
--

test ()

3.2 Variables

Name	Description
I	Value: ' '
__package__	Value: 'gmisclib'

3.3 Class *BadFormatError*



3.3.1 Methods

`__init__(self, t, s)`
`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature
 Overrides: `object.__init__` `exitit`(inherited documentation)

Inherited from `exceptions.ValueError`

`__new__()`

Inherited from `exceptions.BaseException`

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

Inherited from `object`

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

3.3.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
<code>args</code> , <code>message</code>	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

3.4 Class `vec`

object —
 `gmisclib.HTK_HMM_io.vec`

3.4.1 Methods

`__init__(self, vtype, v, name=None)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Parameters

`vtype`: (*type=string, one of 'u', 'v', 'i', 't'*)

Overrides: `object.__init__`

`__eq__(self, other)`

`__hash__(self)`

`hash(x)`

Overrides: `object.__hash__` `exitit`(inherited documentation)

`__repr__(self)`

`repr(x)`

Overrides: `object.__repr__` `exitit`(inherited documentation)

`write(self)`

Inherited from `object`

`__delattr__()`, `__format__()`, `__getattr__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

3.4.2 Properties

Name	Description
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

3.4.3 Class Variables

Name	Description
cname	Value: {'i': 'InvCov', 't': 'TransP', 'u': 'Mean', 'v': 'Variance'}
dim	Value: {'i': -2, 't': 2, 'u': 1, 'v': 1}

3.5 Class mixture

object └─
gmisclib.HTK_HMM_io.mixture

3.5.1 Methods

__init__(*self*, *wt*=1.0, *mean*=None, *var*=None, *name*=None)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

__hash__(*self*)

hash(x)

Overrides: object.__hash__ extit(inherited documentation)

__eq__(*self*, *other*)

check(*self*)

__repr__(*self*)

repr(x)

Overrides: object.__repr__ extit(inherited documentation)

write(*self*, *imix*=None)

Inherited from object

__delattr__(), __format__(), __getattr__(), __new__(), __reduce__(), __reduce_ex__(),
__setattr__(), __sizeof__(), __str__(), __subclasshook__()

3.5.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

3.5.3 Class Variables

Name	Description
<code>code</code>	Value: 'm'

3.6 Class ref

object —
 gmisclib.HTK_HMM_io.ref

3.6.1 Methods

`__init__(self, thing)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` extit(inherited documentation)

`write(self)`

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

3.6.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

3.7 Class state



3.7.1 Methods

```
__init__(self, name=None)
```

x.**__init__**(...) initializes *x*; see *x*.**__class__**.**__doc__** for signature

Overrides: *object*.**__init__** *exitit*(inherited documentation)

```
create_mx(self, i)
```

```
check(self, istate=-1, phone='???')
```

```
__repr__(self)
```

repr(*x*)

Overrides: *object*.**__repr__** *exitit*(inherited documentation)

```
write(self, istate)
```

Inherited from object

__delattr__(), **__format__**(), **__getattr__**(), **__hash__**(), **__new__**(), **__reduce__**(), **__reduce_ex__**(),
__setattr__(), **__sizeof__**(), **__str__**(), **__subclasshook__**()

3.7.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

3.8 Class *hmm*



3.8.1 Methods

```
__init__(self, numstates, name=None)
```

x.**__init__**(...) initializes *x*; see *x*.**__class__**.**__doc__** for signature

Overrides: *object*.**__init__** extit(inherited documentation)

```
check(self)
```

```
write(self)
```

Inherited from object

__delattr__(), **__format__**(), **__getattr__**(), **__hash__**(), **__new__**(), **__reduce__**(), **__reduce_ex__**(),
__repr__(), **__setattr__**(), **__sizeof__**(), **__str__**(), **__subclasshook__**()

3.8.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

3.9 Class *hmmset*



3.9.1 Methods

```
__init__(self)
```

x.**__init__**(...) initializes *x*; see *x*.**__class__**.**__doc__** for signature

Overrides: *object*.**__init__** extit(inherited documentation)

```
twiddle_o(self)
```

```
write(self)
```

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

3.9.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

3.10 Class tokstream

object └─ **gmisclib.HTK_HMM_io.tokstream**

This breaks the HTK HMM file down into tokens.

3.10.1 Methods

<code>__init__(self, fdlist)</code>
<code>x.__init__(...)</code> initializes x; see <code>x.__class__.__doc__</code> for signature
Overrides: <code>object.__init__</code> <code>exitit</code> (inherited documentation)

<code>nextline(self)</code>
Get the next line from the input files(s). Sets <code>self.line</code> .
Return Value
the line.
(<i>type=</i> <code>str</code> or <i>None</i> .)

<code>__nonzero__(self)</code>
Return Value
True when there is more work to be done.

get(*self*)

Get a token. This also sets `self.fulltok` with the full text of the token.

Return Value

The regular expression that matches the next token, and the relevant text.

(*type*=(*compiled regular expression*, *str*))

Raises

BadFormatError When the remainder of the line doesn't match any legal pattern.

get_name(*self*)**get_bracket(*self*)****get_int(*self*)****get_float(*self*)****Inherited from object**

`--delattr--()`, `--format--()`, `--getattr--()`, `--hash--()`, `--new--()`, `--reduce--()`, `--reduce_ex--()`, `--repr--()`, `--setattr--()`, `--sizeof--()`, `--str--()`, `--subclasshook--()`

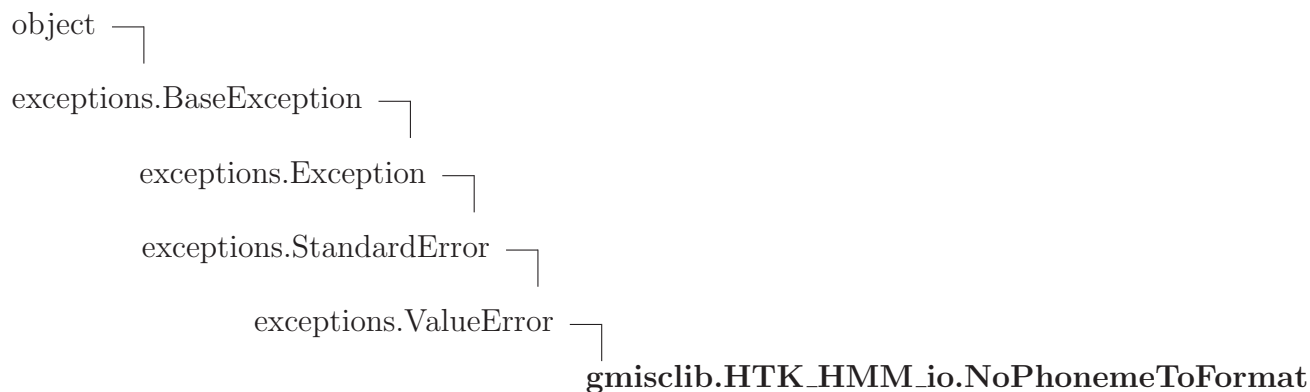
3.10.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

3.10.3 Class Variables

Name	Description
<code>int</code>	Value: <code>re.compile(r'([\+-]?[0-9]+)')</code>
<code>float</code>	Value: <code>re.compile(r'([\+-]?[0-9]*\.[0-9]*[e0-9\+-]*)')</code>
<code>bracket</code>	Value: <code>re.compile(r'<s*(\w+)\s*>')</code>
<code>macro</code>	Value: <code>re.compile(r'~([a-zA-Z])')</code>
<code>name</code>	Value: <code>re.compile(r'"([a-zA-Z_]\w*)"')</code>

3.11 Class *NoPhonemeToFormat*



3.11.1 Methods

`--init--`(*self*, **s*)

x.**`--init--`**(...) initializes *x*; see *x*.**`--class--`**.**`--doc--`** for signature

Overrides: *object*.**`--init--`** *exitit*(inherited documentation)

Inherited from exceptions.ValueError

`--new--`()

Inherited from exceptions.BaseException

`--delattr--`(), **`--getattr--`**(), **`--getitem--`**(), **`--getslice--`**(), **`--reduce--`**(), **`--repr--`**(),
`--setattr--`(), **`--setstate--`**(), **`--str--`**(), **`--unicode--`**()

Inherited from object

`--format--`(), **`--hash--`**(), **`--reduce.ex--`**(), **`--sizeof--`**(), **`--subclasshook--`**()

3.11.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
args, message	
<i>Inherited from object</i>	
<code>--class--</code>	

4 Module **gmisclib.HTK_MLF_io**

This reads MLF (Master Label Files) for/from the HTK speech recognition toolkit.

4.1 Functions

parse_label_line(*s*, *tq*)

This parses a line from a MLF into a tuple.

Parameters

tq: time quantum (normally 1e-7 seconds)

(*type=float*)

s: line to be parsed

(*type=str*)

Raises

BadFileFormatError when parsing is not possible.

readone(*mlf_efn*, *postfix*='.wav', *datapath*='.', *strict*=True, *findfile*=True, *pathedit*=None, *time_quantum*=1e-07, *verbose*=False)

Read a single set of labels from a MLF file. You specify the labels as part of the extended filename, like this: name_of_MLF_file:name_of_labels'. The function returns only a single value and raises an exception if the extended filename is ambiguous.

Parameters

mlf_efn: (*type=string in the form "F:S" @rtype dict()*)

Return Value

a dictionary that describes the labels as per **readiter**.


```
readiter(mlf_fn, postfix='.wav', datapath='.', strict=True, findfile=True,
pathedit=None, time_quantum=1e-07, verbose=False)
```

Read a HTK Master Label (MLF) file. Datapath and pathedit are ways to deal with the situation where the MLF file has been moved, or (for other reasons) the filenames in the MLF file don't point to the actual data.

Parameters

- strict:** If true, raise an exception if an audio file cannot be found.
(*type=bool*)
- time_quantum:** A factor to convert from the time information in the MLF to real units of time (like seconds). Ideally, time_quantum=1e-7 seconds for MLF files, but that isn't exactly accurate for some sampling rates (like 11025 samples/sec) when the sampling interval is not an integral multiple of 100 nanoseconds.
(*type=float*)

Return Value

[{'filespec':path, 'd': d, 'f': f, 'symbols': [...] }, ...]. This is a list (or iterator) of dictionaries. Each dictionary corresponds to one utterance, or one "label file" in the MLF. Attributes 'd' and 'f' are only present if findfile==True; `os.path.join(x['d'], x['f'])` is a path to the corresponding audio. `x['filespec']` is the path information in the MLF, `x['i']` is an `int` indexing which utterance this is within the MLF, and `x['symbols']` is the label information for that utterance. It is a list of tuples produced by `parse_label_line`.

(*type=list of dict (for read) or an iterator producing dicts (for readiter).*)

read(*mlf_fn*, ***kw*)

Read a HTK Master Label (MLF) file. Datapath and pathedit are ways to deal with the situation where the MLF file has been moved, or (for other reasons) the filenames in the MLF file don't point to the actual data.

Parameters

- strict:** If true, raise an exception if an audio file cannot be found.
(*type=bool*)
- time_quantum:** A factor to convert from the time information in the MLF to real units of time (like seconds). Ideally, time_quantum=1e-7 seconds for MLF files, but that isn't exactly accurate for some sampling rates (like 11025 samples/sec) when the sampling interval is not an integral multiple of 100 nanoseconds.
(*type=float*)

Return Value

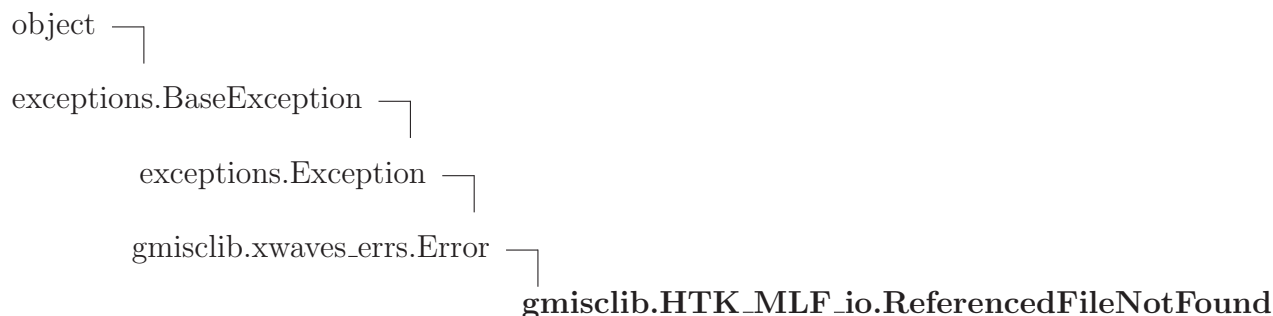
[{'filespec':path, 'd': d, 'f': f, 'symbols': [...] }, ...]. This is a list (or iterator) of dictionaries. Each dictionary corresponds to one utterance, or one "label file" in the MLF. Attributes 'd' and 'f' are only present if findfile==True; `os.path.join(x['d'], x['f'])` is a path to the corresponding audio. `x['filespec']` is the path information in the MLF, `x['i']` is an `int` indexing which utterance this is within the MLF, and `x['symbols']` is the label information for that utterance. It is a list of tuples produced by `parse_label_line`.

(*type=list of dict (for read) or an iterator producing dicts (for readiter).*)

4.2 Variables

Name	Description
TIME_QUANTUM	Value: 1e-07
--package--	Value: 'gmisclib'

4.3 Class *ReferencedFileNotFound*



4.3.1 Methods

`__init__(self, *s)`
`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature
 Overrides: `object.__init__` extit(inherited documentation)

Inherited from exceptions.Exception

`__new__()`

Inherited from exceptions.BaseException

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

Inherited from object

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

4.3.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
<code>args</code> , <code>message</code>	
<i>Inherited from object</i>	
<code>__class__</code>	

4.4 Class writer



4.4.1 Methods

```
__init__(self, mlf_fd, time_quantum=1e-07)
```

x.**__init__**(...) initializes *x*; see *x*.**__class__**.**__doc__** for signature

Overrides: *object*.**__init__** *exitit*(inherited documentation)

```
chunk(self, filespec, data)
```

```
threecol(self, filespec, tdata)
```

```
close(self)
```

```
__del__(self)
```

Inherited from object

__delattr__(), **__format__**(), **__getattr__**(), **__hash__**(), **__new__**(), **__reduce__**(), **__reduce_ex__**(),
__repr__(), **__setattr__**(), **__sizeof__**(), **__str__**(), **__subclasshook__**()

4.4.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

5 Module *gmisclib.MFCCFile*

5.1 Class *MFCCFile*

5.1.1 Methods

<code>__init__(self, filename, DEBUG=0, BYTEORDER='@')</code>

<code>Save(self, filename)</code>

<code>ReturnSent(self, sentnum)</code>
--

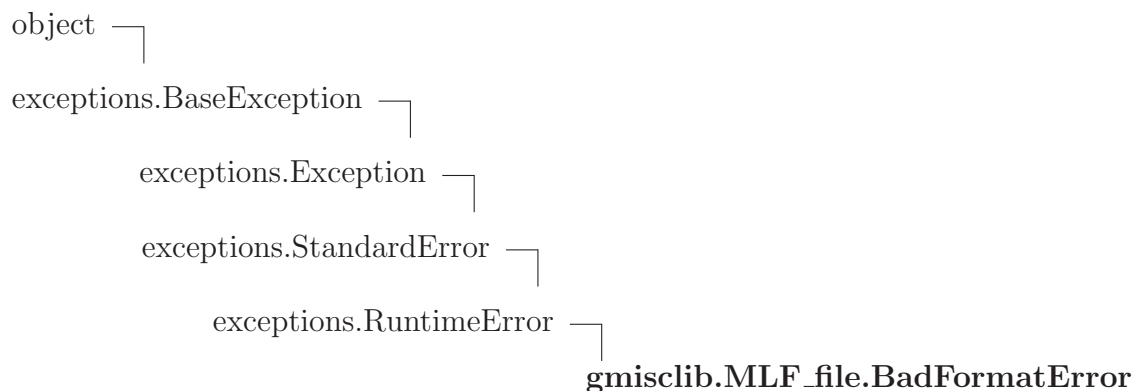
<code>ReturnVector(self, sentnum, vecnum)</code>
--

6 Module *gmisclib.MLF_file*

6.1 Variables

Name	Description
<code>__package__</code>	Value: None

6.2 Class *BadFormatError*



6.2.1 Methods

```

__init__(self, *s)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature
Overrides: object.__init__ extit(inherited documentation)

```

Inherited from exceptions.RuntimeError

`__new__()`

Inherited from exceptions.BaseException

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

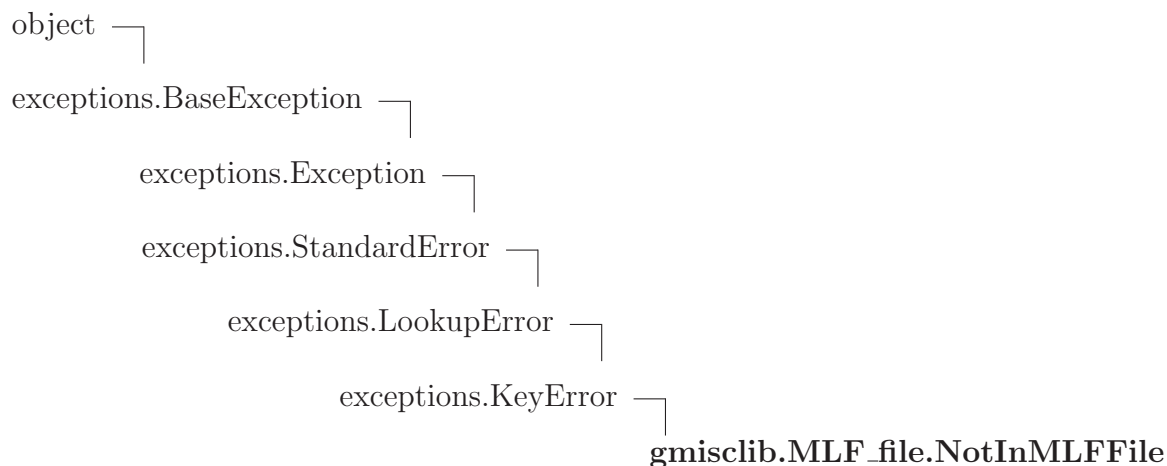
Inherited from object

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

6.2.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
args, message	
<i>Inherited from object</i>	
__class__	

6.3 Class *NotInMLFFile*



6.3.1 Methods

```

__init__(self, *s)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature
Overrides: object.__init__ extit(inherited documentation)

```

Inherited from exceptions.KeyError

```
__new__(), __str__()
```

Inherited from exceptions.BaseException

```

__delattr__(), __getattr__(), __getitem__(), __getslice__(), __reduce__(), __repr__(),
__setattr__(), __setstate__(), __unicode__()

```

Inherited from object

```
__format__(), __hash__(), __reduce_ex__(), __sizeof__(), __subclasshook__()
```

6.3.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
<code>args</code> , <code>message</code>	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

6.4 Class `block_MLF_file`

object └─ `gmisclib.MLF_file.block_MLF_file`

This class reads in and stores a MLF file. It does not interpret the interior data, but rather just breaks it up into blocks, each corresponding to an utterance.

6.4.1 Methods

`__init__(self, fname, preprocessor=None)`

Read in a MLF file and store the information in `self.block`.

Overrides: `object.__init__`

`get(self, key)`

Get a block of text from a MLF file.

`get3(self, key, n=3)`

Get a block of time-aligned labels from a MLF file and interpret it.

Return Value

(`start`, `end`, `label`, ...) tuples, with `start` and `end` in seconds, `label` is a string indicating a phoneme or word or whatever. If there is more information on a line, it will be passed along in the tuple.

(*type*=`list(tuple(start, end, label, ...), ...)`)

Inherited from `object`

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

6.4.2 Properties

Name	Description
<i>Inherited from object</i> __class__	

6.4.3 Class Variables

Name	Description
Quantum	Value: 1e-07

7 Module *gmisclib.Num*

This is a compatibility module, allowing python to work with either *scipy*, *Numeric*, or *numarray*. It tries to import them, in that order, only reporting an error if none are available.

Note that this does not pretend to solve all compatibility problems; it just tries importing all three, so you can only count on the lowest common denominator.

Version: 1.3.0

7.1 Variables

Name	Description
NewAxis	Value: None
ALLOW_THREADS	Value: 1
BUFSIZE	Value: 10000
CLIP	Value: 0
ERR_CALL	Value: 3
ERR_DEFAULT	Value: 0
ERR_DEFAULT2	Value: 2084
ERR_IGNORE	Value: 0
ERR_LOG	Value: 5
ERR_PRINT	Value: 4
ERR_RAISE	Value: 2
ERR_WARN	Value: 1
FLOATING_POINT_SUPPORT	Value: 1
FPE_DIVIDEBYZERO	Value: 1
FPE_INVALID	Value: 8
FPE_OVERFLOW	Value: 2
FPE_UNDERFLOW	Value: 4
False_	Value: False
Inf	Value: inf
Infinity	Value: inf
MAXDIMS	Value: 32
NAN	Value: nan
NINF	Value: -inf
NZERO	Value: -0.0
NaN	Value: nan
PINF	Value: inf
PZERO	Value: 0.0
RAISE	Value: 2
SHIFT_DIVIDEBYZERO	Value: 0

continued on next page

Name	Description
SHIFT_INVALID	Value: 9
SHIFT_OVERFLOW	Value: 3
SHIFT_UNDERFLOW	Value: 6
ScalarType	Value: (<type 'int'>, <type 'float'>, <type 'complex'>, <type '1...
True_	Value: True
UFUNC_BUFSIZE_DEFAULT	Value: 10000
UFUNC_PYVALS_NAME	Value: 'UFUNC_PYVALS'
WRAP	Value: 1
__package__	Value: 'gmisclib'
absolute	Value: <ufunc 'absolute'>
add	Value: <ufunc 'add'>
arccos	Value: <ufunc 'arccos'>
arccosh	Value: <ufunc 'arccosh'>
arcsin	Value: <ufunc 'arcsin'>
arcsinh	Value: <ufunc 'arcsinh'>
arctan	Value: <ufunc 'arctan'>
arctan2	Value: <ufunc 'arctan2'>
arctanh	Value: <ufunc 'arctanh'>
bitwise_and	Value: <ufunc 'bitwise_and'>
bitwise_not	Value: <ufunc 'invert'>
bitwise_or	Value: <ufunc 'bitwise_or'>
bitwise_xor	Value: <ufunc 'bitwise_xor'>
c_	Value: <numpy.lib.index_tricks.CClass object at 0x1739090>
cast	Value: {<type 'numpy.int64'>: <function <lambda> at 0x1627140>, ...
ceil	Value: <ufunc 'ceil'>
conj	Value: <ufunc 'conjugate'>
conjugate	Value: <ufunc 'conjugate'>
cos	Value: <ufunc 'cos'>
cosh	Value: <ufunc 'cosh'>
deg2rad	Value: <ufunc 'deg2rad'>
degrees	Value: <ufunc 'degrees'>
divide	Value: <ufunc 'divide'>
e	Value: 2.71828182846
equal	Value: <ufunc 'equal'>
exp	Value: <ufunc 'exp'>
exp2	Value: <ufunc 'exp2'>
expm1	Value: <ufunc 'expm1'>
fabs	Value: <ufunc 'fabs'>

continued on next page

Name	Description
floor	Value: <ufunc 'floor'>
floor_divide	Value: <ufunc 'floor_divide'>
fmax	Value: <ufunc 'fmax'>
fmin	Value: <ufunc 'fmin'>
fmod	Value: <ufunc 'fmod'>
frexp	Value: <ufunc 'frexp'>
greater	Value: <ufunc 'greater'>
greater_equal	Value: <ufunc 'greater_equal'>
hypot	Value: <ufunc 'hypot'>
index_exp	Value: <numpy.lib.index_tricks.IndexExpression object at 0x1739150>
inf	Value: inf
infty	Value: inf
invert	Value: <ufunc 'invert'>
isfinite	Value: <ufunc 'isfinite'>
isinf	Value: <ufunc 'isinf'>
isnan	Value: <ufunc 'isnan'>
ldexp	Value: <ufunc 'ldexp'>
left_shift	Value: <ufunc 'left_shift'>
less	Value: <ufunc 'less'>
less_equal	Value: <ufunc 'less_equal'>
little_endian	Value: True
log	Value: <ufunc 'log'>
log10	Value: <ufunc 'log10'>
log1p	Value: <ufunc 'log1p'>
logaddexp	Value: <ufunc 'logaddexp'>
logaddexp2	Value: <ufunc 'logaddexp2'>
logical_and	Value: <ufunc 'logical_and'>
logical_not	Value: <ufunc 'logical_not'>
logical_or	Value: <ufunc 'logical_or'>
logical_xor	Value: <ufunc 'logical_xor'>
maximum	Value: <ufunc 'maximum'>
mgrid	Value: <numpy.lib.index_tricks.nd_grid object at 0x16ece90>
minimum	Value: <ufunc 'minimum'>
mod	Value: <ufunc 'remainder'>
modf	Value: <ufunc 'modf'>
multiply	Value: <ufunc 'multiply'>
nan	Value: nan
nbytes	Value: {<type 'numpy.int64'>: 8, <type 'numpy.int16'>: 2, <type ...

continued on next page

Name	Description
<code>negative</code>	Value: <code><ufunc 'negative'></code>
<code>newaxis</code>	Value: <code>None</code>
<code>not_equal</code>	Value: <code><ufunc 'not_equal'></code>
<code>ogrid</code>	Value: <code><numpy.lib.index_tricks.nd_grid object at 0x16eced0></code>
<code>ones_like</code>	Value: <code><ufunc 'ones_like'></code>
<code>pi</code>	Value: <code>3.14159265359</code>
<code>power</code>	Value: <code><ufunc 'power'></code>
<code>r_</code>	Value: <code><numpy.lib.index_tricks.RClass object at 0x16ecf90></code>
<code>rad2deg</code>	Value: <code><ufunc 'rad2deg'></code>
<code>radians</code>	Value: <code><ufunc 'radians'></code>
<code>reciprocal</code>	Value: <code><ufunc 'reciprocal'></code>
<code>remainder</code>	Value: <code><ufunc 'remainder'></code>
<code>right_shift</code>	Value: <code><ufunc 'right_shift'></code>
<code>rint</code>	Value: <code><ufunc 'rint'></code>
<code>s_</code>	Value: <code><numpy.lib.index_tricks.IndexExpression object at 0x17391d0></code>
<code>sctypeDict</code>	Value: <code>{0: <type 'numpy.bool_'>, 1: <type 'numpy.int8'>, 2: <typ...</code>
<code>sctypeNA</code>	Value: <code>{'?': 'Bool', 'B': 'UInt8', 'Bool': <type 'numpy.bool_'>, ...</code>
<code>sctypes</code>	Value: <code>{'complex': [<type 'numpy.complex64'>, <type 'numpy.compl...</code>
<code>sign</code>	Value: <code><ufunc 'sign'></code>
<code>signbit</code>	Value: <code><ufunc 'signbit'></code>
<code>sin</code>	Value: <code><ufunc 'sin'></code>
<code>sinh</code>	Value: <code><ufunc 'sinh'></code>
<code>sqrt</code>	Value: <code><ufunc 'sqrt'></code>
<code>square</code>	Value: <code><ufunc 'square'></code>
<code>subtract</code>	Value: <code><ufunc 'subtract'></code>
<code>tan</code>	Value: <code><ufunc 'tan'></code>
<code>tanh</code>	Value: <code><ufunc 'tanh'></code>
<code>true_divide</code>	Value: <code><ufunc 'true_divide'></code>
<code>trunc</code>	Value: <code><ufunc 'trunc'></code>
<code>typeDict</code>	Value: <code>{0: <type 'numpy.bool_'>, 1: <type 'numpy.int8'>, 2: <typ...</code>
<code>typeNA</code>	Value: <code>{'?': 'Bool', 'B': 'UInt8', 'Bool': <type 'numpy.bool_'>, ...</code>

continued on next page

Name	Description
typecodes	Value: {'All': '?bhilqpBHILQPfdgFDGSUV0', 'AllFloat': 'fdgFDG', ...}

8 Module *gmisclib.Numeric_gpk*

8.1 Functions

add_overlap(*a*, *astart*, *b*, *bstart*)

Add arrays *a* and *b* in the overlap region. Return (*data*, *start*). If *a*, *b* are time series, they are assumed to have the same sampling rate. *Astart* and *Bstart* apply to the zeroth index. All other indices are assumed to match start and length.

zero_pad_end(*d*, *padfactor*=1)

Poisson(*nbar*)

Return a Poisson random integer, whose distribution has a mean = *nbar*.

bevel_concat(*a*, *b*, *bevel*=0, *bevel_overlap*=1.0, *delay*=0, *ta*=None, *tb*=None)

Concatenate two time series. Bevel the edges, and overlap them slightly.

Bevel_overlap controls the fractional overlap of the two bevels, and *delay* specifies an extra delay for *b*.

If *ta* and/or *tb* are specified, return a tuple of (*concatenated_time_series*, *tma*, *tmb*) where *tma* and *tmb* are the locations corresponding to *ta* and *tb* in the corresponding input arrays.

argmax(*a*)

N_maximum(*a*)

N_minimum(*a*)

N_frac_rank(*a*, *fr*)

N_mean_ad(*a*)

Mean absolute deviation. For a multi-dimensional array, it takes the MAD along the first axis, so `N_mean_ad(x)[0]==N_mean_ad(x[:,0])`.

median_across(*list_of_vec*)

Returns an element-by-element median of a list of Numeric vectors.

N_median(*a*, *axis*=0)

Returns an element-by-element median of a list of Numeric vectors.

N_median_across(*a*)

set_diag(*x*, *a*)

Set the diagonal of a matrix *x* to be the vector *a*. If *a* is shorter than the diagonal of *x*, just set the beginning.

limit(*low*, *x*, *high*)

trimmed_mean_sigma_across(*list_of_vec*, *weights*, *clip*)

trimmed_mean_across(*list_of_vec*, *weights*, *clip*)

trimmed_stdev_across(*list_of_vec*, *weights*, *clip*)

vec_variance(*x*)

Take a component-by-component variance of a list of vectors.

qform(*vec*, *mat*)

A quadratic form: *vec***mat***vec*, or *vecs***mat***transpose*(*vecs*)

KolmogorovSmirnov(*d1*, *d2*, *w1*=None, *w2*=None)

interpN(*a*, *t*)

Interpolate array *a* to floating point indices, *t*, via nearest-neighbor interpolation. Returns a Numeric array.

interp(*a*, *t*)

Interpolate to a specified time axis. This does a linear interpolation. *A* is a Numpy array, and *t* is an array of times.

Return Value

interpolated values

(*type*=*numpy array*.)

split_into_clumps(*x*, *threshold*, *minsize*=1)

This reports when the signal is above the threshold.

Parameters

x: a signal
(*type*=*numpy.ndarray*, *one-dimensional*.)

threshold: a threshold.
(*type*=*float*)

Return Value

[(start, stop), ...] for each region ("clump") where **x**>**threshold**.

block_stdev(*x*)

This is just a alternative implementation of the standard deviation of each channel, but it is designed in a block-wise fashion so the total memory usage is not large.

convolve(*x*, *kernel*)

This is basically like `numpy.convolve(x, kernel, 1)` except that it properly handles the case where the kernel is longer than the data.

```
edge_window(n, eleft, eright, typeleft='linear', typeright='linear',  
norm=None)
```

Creates a window which is basically flat, but tapers off on the left and right edges. The widths of the tapers can be controlled, as can the shapes.

Parameters

- n:** the width of the window
(*type=int*)
- eleft:** the width of the left taper (i.e. at zero index, in samples).
(*type=int*)
- eright:** the width of the right taper (i.e. at index near **n**, in samples).
(*type=int*)
- typeleft:** what kind of taper on the left? (Defaults to "linear").
(*type='linear' or 'cos'*)
- typeright:** what kind of taper on the right? (Defaults to "linear").
(*type='linear' or 'cos'*)
- norm:** How to normalize? The default is **None**, which means no normalization. Providing a number **x** will normalize the window so that the average of the **window**x==1**.
(*type=None or float != 0.*)

```
edge_window_t(t, eleft, eright, typeleft='linear', typeright='linear',  
norm=None)
```

Computes a window which is basically flat, but tapers off on the left and right edges. The widths of the tapers can be controlled, as can the shapes.

Parameters

t: an array of time values. They are required to be monotonically increasing, and assumed to be linearly spaced.
(*type=*`numpy.ndarray`)

eleft: the width of the left taper (i.e. at zero index, in time units).
(*type=*`float`)

eright: the width of the right taper (i.e. at index near *n*, in time units).
(*type=*`float`)

typeleft: what kind of taper on the left? (Defaults to "linear").
(*type=*`'linear'` or `'cos'`)

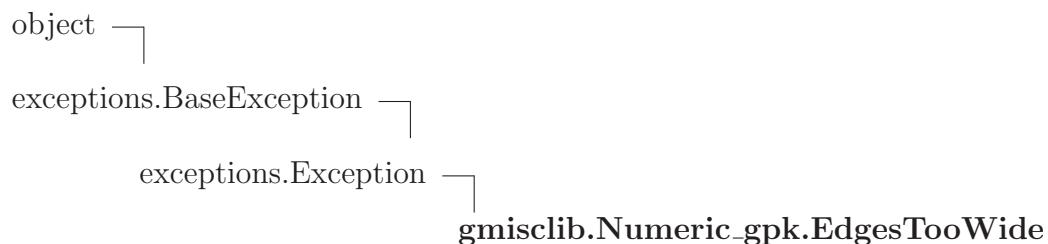
typeright: what kind of taper on the right? (Defaults to "linear").
(*type=*`'linear'` or `'cos'`)

norm: How to normalize? The default is `None`, which means no normalization. Providing a number *x* will normalize the window so that the average of the `window**x==1`.
(*type=*`None` or `float != 0`.)

8.2 Variables

Name	Description
<code>pylab</code>	Value: <code>None</code>
<code>asinh</code>	Value: <code><ufunc 'arcsinh'></code>
<code>acosh</code>	Value: <code><ufunc 'arccosh'></code>
<code>BLOCK</code>	Value: <code>8192</code>
<code>--package--</code>	Value: <code>'gmisclib'</code>

8.3 Class *EdgesTooWide*



8.3.1 Methods

`--init--`(*self*, **s*)

x.**`--init--`**(...) initializes *x*; see *x*.**`--class--`**.**`--doc--`** for signature

Overrides: *object*.**`--init--`** *exitit*(inherited documentation)

Inherited from exceptions.Exception

`--new--`()

Inherited from exceptions.BaseException

`--delattr--`(), **`--getattr--`**(), **`--getitem--`**(), **`--getslice--`**(), **`--reduce--`**(), **`--repr--`**(),
`--setattr--`(), **`--setstate--`**(), **`--str--`**(), **`--unicode--`**()

Inherited from object

`--format--`(), **`--hash--`**(), **`--reduce_ex--`**(), **`--sizeof--`**(), **`--subclasshook--`**()

8.3.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
<i>args</i> , <i>message</i>	
<i>Inherited from object</i>	
<i>--class--</i>	

9 Module `gmisclib.accent_spec`

This module provides a way of safely specifying accent positions in running text.

If you have a transcription "I did not eat the orange ball.", you can attach a "+" symbol to the word "eat" like this: "+eat".

You enter an accent specification, which consists of words with a prefix (the prefix is anything that ends in a punctuation mark). The program matches up the words in the accent specification to the words in the transcription. You can have many words in the accent specification, if necessary. You can also disambiguate things by putting context words into the accent spec (without a prefix). All matching is done left-to-right.

Version: \$Revision: 1.4 \$

9.1 Functions

```
prefix(text_array, accent_spec, map_fcn=<function <lambda> at
0x1293758>)
```

This function takes an array of words and an accent spec. It matches the accent spec to the words, and outputs an array of tuples which tells you where the accents are, and what kind. The optional `map_fcn` can be used to map other kinds of objects into a array of strings.

More specifically, an `accent_spec` is a whitespace-separated list of strings. Each string is a word from the `text_array`, with an optional prefix. The strings are matched in order to the words, and the output array is a list of (index_in_text_array, prefix_text) tuples.

So, if you have `text_array = ['my', 'cat', 'is', 'my', 'cat']` and `accent_spec="is +my"`, then `align()` will match "is" to `text_array[2]`, and "+my" to `text_array[3]`, and it will return `[(3, "+")]`. Note that "+is +my" does not imply that 'is' and 'my' are adjacent, and "+is +cat" simply returns `[(2, "+"), (4, "+")]`.

If the `accent_spec` were "+my", then it would match `text_array[0]`, and return `[(0, "+")]`.

Prefixes can be multiple characters, but they cannot end in letters, digits, or underscore.

```
suffix(text_array, accent_spec, map_fcn=<function <lambda> at
0x12938c0>)
```

See `prefix`, but with the obvious changes.

```
preshow(text_array, alignment, map_fcn=<function <lambda> at 0x12939b0>)
```

Shows an alignment in a printable form. It puts the prefixes in the appropriate places.

```
sufshow(text_array, alignment, map_fcn=<function <lambda> at 0x1293aa0>)
```

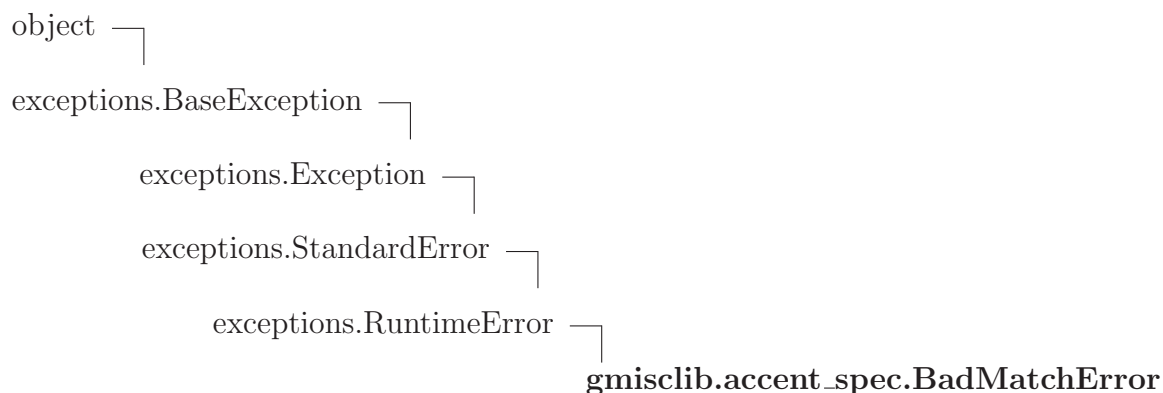
Shows an alignment in a printable form. It puts the suffixes in the appropriate places.

```
test()
```

9.2 Variables

Name	Description
<code>__package__</code>	Value: <code>'gmisclib'</code>

9.3 Class `BadMatchError`



9.3.1 Methods

```
__init__(self, x)
```

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` extit(inherited documentation)

Inherited from `exceptions.RuntimeError`

`__new__()`

Inherited from exceptions.BaseException

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

Inherited from object

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

9.3.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
args, message	
<i>Inherited from object</i>	
__class__	

10 Module `gmisclib.array_window`

Defines `array_window` class.

Version: \$Revision: 1.4 \$

10.1 Functions

<code>test()</code>

10.2 Variables

Name	Description
<code>--package--</code>	Value: None

10.3 Class `array_window`

Lets you reference 1-d array items relative to an arbitrary zero.

10.3.1 Methods

<code>--init--(<i>self</i>, <i>array</i>, <i>zero</i>)</code>

<code>--setattr--(<i>self</i>, <i>name</i>, <i>value</i>)</code>
--

<code>--delattr--(<i>self</i>, <i>name</i>)</code>
--

<code>--getitem--(<i>self</i>, <i>index</i>)</code>

<code>--setitem--(<i>self</i>, <i>index</i>, <i>value</i>)</code>

<code>--getslice--(<i>self</i>, <i>i</i>, <i>j</i>)</code>
--

<code>--setslice--(<i>self</i>, <i>i</i>, <i>j</i>, <i>sequence</i>)</code>

<code>--str--(<i>self</i>)</code>

<code>--repr--(self)</code>

<code>--len--(self)</code>

Of course, one can't assume that one can say <code>x[i]</code> for <code>i</code> in <code>range(len(x))</code> .

10.3.2 Class Variables

Name	Description
<code>--doc--</code>	Value: <code>"""Lets you reference 1-d array items relative to an arbi...</code>

11 Module *gmisclib.avio*

Module to parse and create lines in "a=v;" format. Normally, the entry points are `parse()` and `concoct()`.

Version: \$Revision: 1.21 \$

11.1 Functions

parse(*s*, *sep*=';')

Parses a line in a=v; a=v form into a dictionary. The line may be terminated by a '#' followed by an arbitrary comments. Both attributes and values are assumed to be encoded with the default `g_encode` encoder. (However, the comment is not assumed to be encoded.)

Parameters

s: the string to be parsed.
sep: (optional) lets you choose the character that separates attribute-value pairs.

Return Value

dictionary containing a=v pairs. One special entry for comments: "_COMMENT".

Raises

`BadFormatError` when *s* is not formatted properly.

concoct(*s*)

Converts a dictionary to a line in the form a=v;a=v;#comment. It returns a string. The entries in the dictionary will be converted to a string representation by `fwd()`.

read(*fd*, *sep*='; ', *line*=None)

Read a file in a=v; format. If *line* is set, put the line number into a slot with the specified name.

Return Value

(data, comments) where data=[{a:v, ...}, ...] and comments=[str, ...]
(type=tuple(list(dict), list(str)))

Raises

`BadFormatError` when *s* is not formatted properly.

test1()

read_hdc(*fd*, *sep*=';', *line*=None)

This emulates `fiatio.read()`.

Return Value

(header, data, comments) where data=[{a:v, ...}, ...] and
comments=[str, ...]

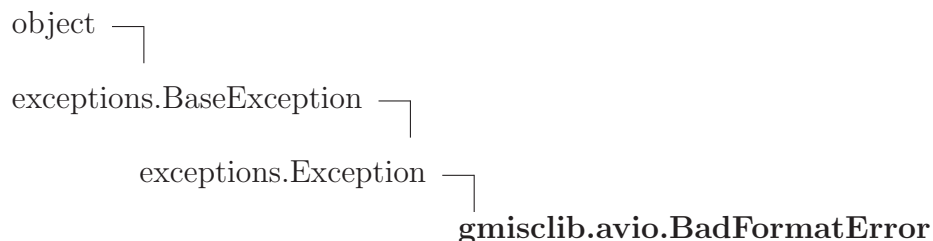
(*type=tuple(dict, list(dict), list(str))*)

test2()

11.2 Variables

Name	Description
<code>--package--</code>	Value: 'gmisclib'

11.3 Class `BadFormatError`



11.3.1 Methods

__init__(*self*, **x*)

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` `extit`(inherited documentation)

Inherited from `exceptions.Exception`

`__new__`()

Inherited from `exceptions.BaseException`

`__delattr__`(), `__getattr__`(), `__getitem__`(), `__getslice__`(), `__reduce__`(), `__repr__`(),

`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

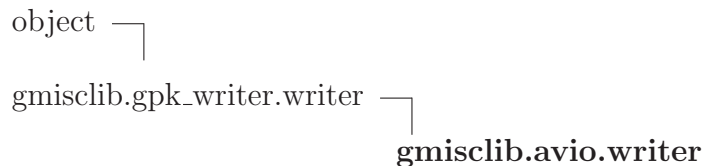
Inherited from object

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

11.3.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
args, message	
<i>Inherited from object</i>	
<code>__class__</code>	

11.4 Class writer



Write a file in a=v; format.

11.4.1 Methods

comment (<i>self</i> , <i>comment</i>) <hr/> Add a comment to the data file. Overrides: gmisclib.gpk_writer.writer.comment
header (<i>self</i> , <i>k</i> , <i>v</i>) <hr/> NOTE: this is not a fully general function. Keys may not have spaces or equals signs in them. Note also that reading of headers from a=v; files is not supported. Overrides: gmisclib.gpk_writer.writer.header
__init__ (<i>self</i> , <i>fd</i>) <hr/> <i>x.__init__(...)</i> initializes <i>x</i> ; see <i>x.__class__.__doc__</i> for signature Overrides: object.__init__ <code>exitit</code> (inherited documentation)

datum (<i>self</i> , <i>data_item</i>)

Overrides: gmisclib.gpk_writer.writer.datum

append (<i>self</i> , <i>d</i>)
--

close (<i>self</i>)

comments (<i>self</i> , <i>comments</i>)

Add comments to the data file. Comments can appear anywhere.
--

data (<i>self</i> , <i>dataset</i>)
--

Write a series of lines to the output file.

extend (<i>self</i> , <i>d</i>)
--

flush (<i>self</i>)

headers (<i>self</i> , <i>h</i>)

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

11.4.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

11.4.3 Class Variables

Name	Description
<code>__doc__</code>	Value: <code>""Write a file in a=v; form...</code>

12 Module *gmisclib.bark_scale*

Version: \$Revision: 1.3 \$

12.1 Functions

asinh (x)

acosh (x)

f_to_bark (f)

frequency to critical band number. See Schroeder et al. (1979).

bark_to_f (b)

critical band number -> frequency M. R. Schroeder, B.S. Atal, J.L. Hall (1979), J.Acoust.Soc.Am. 66(6) 1647-1652. Title: "Optimizing Digital Speech Coders by exploiting masking properties of the Human Ear."
--

cbw (b)

Critical bandwidth (in Hz) at frequency b (barks).
--

12.2 Variables

Name	Description
<code>--package--</code>	Value: 'gmisclib'

13 Module *gmisclib.beamsearch*

13.1 Functions

beamsearch(*cost*, *extra*, *initial*, *B*, *E*)

A breadth-first beam search. B = max number of options to keep, E = max cost difference between best and worst threads in beam. initial = [starting positions] extra = arbitrary information for cost function. cost = fn(state, extra) -> (total_cost, [next states], output_if_goal)

13.2 Variables

Name	Description
--package--	Value: None

14 Module *gmisclib.blue_data_attributes*

This chooses samples such that the specified attributes are broadly distributed.

14.1 Functions

test()

14.2 Variables

Name	Description
<code>__package__</code>	Value: <code>'gmisclib'</code>

14.3 Class *blue_attributes*

object  `gmisclib.blue_data_attributes.blue_attributes`

14.3.1 Methods

<code>__init__(self, inspector, data)</code>
<code>x.__init__(...)</code> initializes x; see <code>x.__class__.__doc__</code> for signature
Parameters
<code>inspector</code> : a function on a datum that returns a list of attributes.
Overrides: <code>object.__init__</code>

<code>pick(self, n)</code>

<code>pick_one(self)</code>

Pick one item from the data set.

peek(*self*)

Inspect (but do not remove) the next item to be picked.

Return Value

the next item to be picked.

(*type=whatever (not a list!)*)

add(*self*, *datum*)

Add another datum to be sampled.

Parameters

datum: thing to be added. It has a probability of $1/\text{len}(\text{self})$ of being the next sample.

(*type=whatever*)

__len__(*self*)

reset(*self*)

Forget prior history of usage. Choices after this call are uncorrelated with choices before this call.

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

14.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

15 Module `gmisclib.blue_data_selector`

This takes anticorrelated samples from a sequence of data. In other words, it tends to choose data that have not been seen frequently before. With `blueness=1`, it never shows a sample `n` times until all samples have been shown `n-1` times.

(Note that it is possible for an item to appear twice in succession, if it appears as the `N`th item in a set of `N`, then appears as the first item in the second set of `N`.)

15.1 Functions

<code>test()</code>

15.2 Variables

Name	Description
<code>--package--</code>	Value: <code>'gmisclib'</code>

15.3 Class `bluedata`

object  `gmisclib.blue_data_selector.bluedata`

15.3.1 Methods

__init__(*self*, *data*, *blueness*=2.0, *rng*=None)

x.__init__(...) initializes *x*; see *x.__class__.__doc__* for signature

Parameters

data: data to sample

(*type=list(whatever)*)

blueness: How much should you avoid choosing the same item in succession? If $-0.5 < \text{blueness} < 0$, then you are more likely to choose the same item several times in a row. If $0 < \text{blueness} < 1$, you are unlikely to choose the same item twice in succession (though it is possible). If $\text{blueness} > 1$, you will never get the same item twice until you've run through the entire list of data.

(*type=float*)

rng: a random number generator, per the **random** module. You can generate repeated sequences of data by repeatedly passing it a random number generator in the same state.

Overrides: *object.__init__*

add(*self*, *datum*)

Add another datum to be sampled.

Parameters

datum: thing to be added. It has a probability of $1/\text{len}(\text{self})$ of being the next sample.

(*type=whatever*)

pickiter(*self*, *n*)

Pick *n* items from the data set.

Parameters

n: how many items to pick.

(*type=int*)

pick1(*self*)

pick(*self*, *n*)Pick *n* items from the data set.**Parameters***n*: how many items to pick.*(type=int)***Return Value**list(whatever), with length==*n***peek**(*self*)

Inspect (but do not remove) the next item to be picked.

Return Value

the next item to be picked.

*(type=whatever (not a list!))***split**(*self*, *n*)

Split the data set into two parts, of size *n* and `len(data)-n`. Successive calls to `split` will avoid having the same items appear in the first part. For instance, given a `bluedata` of size 100, with ten calls to `split(10)`, each datum will appear in the first part of the return tuple exactly once.

Return Value

tuple(list, list)

__iter__(*self*)

This iterator will produce samples forever.

__len__(*self*)**reset**(*self*)

Forget prior history of usage. Choices after this call are uncorrelated with choices before this call.

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

15.3.2 Properties

Name	Description
<i>Inherited from object</i> __class__	

16 Module gmisclib.cache

16.1 Functions

cachepath(*f*, *suffix_to_del*=None, *tail*='', *root*='')

Return a pathname suitable for cacheing some result.

Parameters

f: An arbitrary key, could be a pathname.
(*type*=*str*)

suffix_to_del: something to delete from the tail of **f**. (Or none, if nothing is to be deleted.)
(*type*=*str*)

tail: something to add at the end of the constructed path.
(*type*=*str* or *None*)

Return Value

(*path_to_root*, *path_with_tail*). *Path_to_root* is/will be a directory; *path_with_tail* is a path to a data file within that directory.
(*type*=*tuple*(*str*, *str*))

Raises

ValueError if *suffix_to_del* is specified and **f** doesn't end that way.

16.2 Variables

Name	Description
<code>__package__</code>	Value: 'gmisclib'

17 Module *gmisclib.chunkio*

These are I/O routines to allow you to write stuff like arrays and dictionaries (and arrays of dictionaries) to a human-readable file.

The format is normally STARTMARKER LENGTH.INFO DATA ENDMARKER, where STARTMARKER is something like "a{}", that specifies you're at the beginning of an array. LENGTH.INFO can depend on the data type, but it's normally an integer. Then comes the data, and finally as a check, you encounter the ENDMARKER, which is normally "". All of this is recursive, of course.

Version: \$Revision: 1.24 \$

17.1 Functions

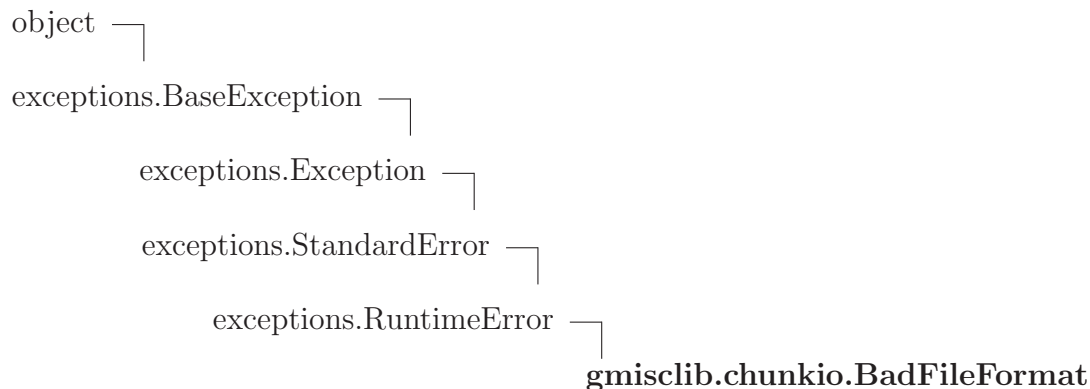
<code>test_e()</code>

<code>test()</code>

17.2 Variables

Name	Description
<code>--package--</code>	Value: 'gmisclib'

17.3 Class *BadFileFormat*



17.3.1 Methods

`__init__(self, s)`
`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature
 Overrides: `object.__init__` extit(inherited documentation)

Inherited from exceptions.RuntimeError`__new__()`***Inherited from exceptions.BaseException***

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

Inherited from object

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

17.3.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
<code>args</code> , <code>message</code>	
<i>Inherited from object</i>	
<code>__class__</code>	

17.4 Class *chunk***Known Subclasses:** `gmislib.chunkio.datachunk`, `gmislib.chunkio.stringchunk`

Low level file I/O operations. This class represents a sequence of white-space separated chunks of data.

17.4.1 Methods

`__init__(self)`
 Constructor.

`more(self)`
 Returns zero if the data source is empty. Returns nonzero if there is more data.

readchunk (<i>self</i>)

Read in the next white-space delimited chunk of text.

read_float (<i>self</i>)

read_array (<i>self</i> , <i>fcn</i>)
--

Read an array of data values. Raw text is converted to finished array values by the specified fcn.
--

read_tuple (<i>self</i> , <i>fcn</i>)
--

Read a tuple of data values. Raw text is converted to finished array values by the specified fcn.

read_array_of (<i>self</i> , <i>fcn</i>)

Read an array of data values. Values are read in by the specified fcn.
--

read_dict (<i>self</i> , <i>fcn</i>)

Read a dictionary of data values. Raw text is converted to finished values by the specified fcn. Keys are strings.
--

read_dict_of (<i>self</i> , <i>fcn</i>)
--

Read a dictionary of data values. Values are read in by the specified fcn. This allows dictionaries of X, where X can be a complex datatype like an array. Keys are strings.
--

groupstart (<i>self</i>)

groupend (<i>self</i>)

read_NumArray (<i>self</i>)

17.4.2 Class Variables

Name	Description
<code>__doc__</code>	Value: <code>""</code> "Low level file I/O operatio..."

17.5 Class *datachunk*

gmisclib.chunkio.chunk └─ **gmisclib.chunkio.datachunk**

Low level file I/O operations. This class represents a file as a sequence of white-space separated chunks of data.

17.5.1 Methods

<code>__init__(self, fd)</code>
Constructor. Overrides: gmisclib.chunkio.chunk.__init__
<code>more(self)</code>
Returns False if the data source is empty. Return Value True if there is more data. Overrides: gmisclib.chunkio.chunk.more
<code>readchunk(self)</code>
Read in the next white-space delimited chunk of text. Overrides: gmisclib.chunkio.chunk.readchunk
<code>groupend(self)</code>
<code>groupstart(self)</code>
<code>read_NumArray(self)</code>
<code>read_array(self, fcn)</code>
Read an array of data values. Raw text is converted to finished array values by the specified fcn.
<code>read_array_of(self, fcn)</code>
Read an array of data values. Values are read in by the specified fcn.

read_dict(*self*, *fcn*)

Read a dictionary of data values. Raw text is converted to finished values by the specified fcn. Keys are strings.

read_dict_of(*self*, *fcn*)

Read a dictionary of data values. Values are read in by the specified fcn. This allows dictionaries of X, where X can be a complex datatype like an array. Keys are strings.

read_float(*self*)

read_tuple(*self*, *fcn*)

Read a tuple of data values. Raw text is converted to finished array values by the specified fcn.

17.5.2 Class Variables

Name	Description
<code>__doc__</code>	Value: <code>""</code> "Low level file I/O operatio..."

17.6 Class *stringchunk*

gmisclib.chunkio.chunk └─ **gmisclib.chunkio.stringchunk**

Low level operations: splitting a string into chunks. This class represents a string as a sequence of white-space separated chunks of data.

17.6.1 Methods

__init__(*self*, *s*)

Constructor.

Overrides: gmisclib.chunkio.chunk.__init__

more(*self*)

Returns zero if the data source is empty. Returns nonzero if there is more data.

Return Value

zero if the data source is empty; nonzero if there is more data.

Overrides: *gmisclib.chunkio.chunk.more*

readchunk(*self*)

Read in the next white-space delimited chunk of text.

Overrides: *gmisclib.chunkio.chunk.readchunk*

groupend(*self*)**groupstart**(*self*)**read_NumArray**(*self*)**read_array**(*self*, *fcn*)

Read an array of data values. Raw text is converted to finished array values by the specified *fcn*.

read_array_of(*self*, *fcn*)

Read an array of data values. Values are read in by the specified *fcn*.

read_dict(*self*, *fcn*)

Read a dictionary of data values. Raw text is converted to finished values by the specified *fcn*. Keys are strings.

read_dict_of(*self*, *fcn*)

Read a dictionary of data values. Values are read in by the specified *fcn*. This allows dictionaries of X, where X can be a complex datatype like an array. Keys are strings.

read_float(*self*)**read_tuple**(*self*, *fcn*)

Read a tuple of data values. Raw text is converted to finished array values by the specified *fcn*.

17.6.2 Class Variables

Name	Description
<code>__doc__</code>	Value: <code>"""Low level operations: splitting a string into chun...</code>

17.7 Class *chunk_w*

Known Subclasses: *gmisclib.chunkio.chunkstring_w*, *gmisclib.chunkio.datachunk_w*

17.7.1 Methods

```
__init__(self)
```

```
writetchunk(self, ch, b=0)
```

```
write(self, ch, b=0)
```

```
nl(self)
```

```
comment(self, comment)
```

```
close(self)
```

```
write_None(self)
```

```
write_array_of(self, data, writer, b=0)
```

```
write_array(self, data, b=0, converter=<type 'str'>)
```

```
write_dict_of(self, data, writer, b=1)
```

```
write_dict(self, data, b=0, converter=<type 'str'>)
```

```
groupstart(self, contents, b=1, comment=None)
```

The 'contents' argument is conventionally a string containing 'a' for an internal array, 'g' for a group, 'd' for a dictionary... It describes the contents of the group. This is just used by the application to check what is in the group, so it could contain any chunk.

```
groupend(self, b=1)
```

```
write_NumArray(self, d, b=0, converter=<type 'str'>)
```

```
write_float(self, d, b=0)
```

```
stringwrite(self, ch, b=0)
```

17.8 Class `datachunk_w`

```
gmisclib.chunkio.chunk_w └─ gmisclib.chunkio.datachunk_w
```

This writes stuff to a file.

17.8.1 Methods

```
__init__(self, fd, width=80)
```

Overrides: `gmisclib.chunkio.chunk_w.__init__`

```
stringwrite(self, ch, b=0)
```

ch is the chunk of text. *b*=1 to begin a new line.

Overrides: `gmisclib.chunkio.chunk_w.stringwrite`

```
nl(self)
```

Overrides: `gmisclib.chunkio.chunk_w.nl`

```
comment(self, comment)
```

Overrides: `gmisclib.chunkio.chunk_w.comment`

close(*self*)Overrides: *gmisclib.chunkio.chunk_w.close***__del__**(*self*)**groupend**(*self*, *b=1*)**groupstart**(*self*, *contents*, *b=1*, *comment=None*)

The 'contents' argument is conventionally a string containing 'a' for an internal array, 'g' for a group, 'd' for a dictionary... It describes the contents of the group. This is just used by the application to check what is in the group, so it could contain any chunk.

write(*self*, *ch*, *b=0*)**write_None**(*self*)**write_NumArray**(*self*, *d*, *b=0*, *converter=<type 'str'>*)**write_array**(*self*, *data*, *b=0*, *converter=<type 'str'>*)**write_array_of**(*self*, *data*, *writer*, *b=0*)**write_dict**(*self*, *data*, *b=0*, *converter=<type 'str'>*)**write_dict_of**(*self*, *data*, *writer*, *b=1*)**write_float**(*self*, *d*, *b=0*)**writetechunk**(*self*, *ch*, *b=0*)

17.8.2 Class Variables

Name	Description
<code>__doc__</code>	Value: <code>"""This writes stuff to a file."""</code>

17.9 Class `chunkstring_w`

`gmisclib.chunkio.chunk_w` └─ **`gmisclib.chunkio.chunkstring_w`**

This accumulates stuff in memory, and returns a string when `close()` is called.

17.9.1 Methods

`__init__(self)`

Overrides: `gmisclib.chunkio.chunk_w.__init__`

`stringwrite(self, ch, b=0)`

Overrides: `gmisclib.chunkio.chunk_w.stringwrite`

`nl(self)`

Overrides: `gmisclib.chunkio.chunk_w.nl`

`comment(self, comment)`

Overrides: `gmisclib.chunkio.chunk_w.comment`

`close(self)`

Overrides: `gmisclib.chunkio.chunk_w.close`

`groupend(self, b=1)`

`groupstart(self, contents, b=1, comment=None)`

The 'contents' argument is conventionally a string containing 'a' for an internal array, 'g' for a group, 'd' for a dictionary... It describes the contents of the group. This is just used by the application to check what is in the group, so it could contain any chunk.

`write(self, ch, b=0)`

`write_None(self)`

`write_NumArray(self, d, b=0, converter=<type 'str'>)`


```
write_array(self, data, b=0, converter=<type 'str'>)
```

```
write_array_of(self, data, writer, b=0)
```

```
write_dict(self, data, b=0, converter=<type 'str'>)
```

```
write_dict_of(self, data, writer, b=1)
```

```
write_float(self, d, b=0)
```

```
writetechunk(self, ch, b=0)
```

17.9.2 Class Variables

Name	Description
<code>__doc__</code>	Value: <code>""</code> This accumulates stuff in memory, and returns a str...

18 Module `gmisclib.convex_hull2d`

`convexhull.py`

Calculate the convex hull of a set of n 2D-points in $O(n \log n)$ time. Taken from Berg et al., Computational Geometry, Springer-Verlag, 1997. Prints output as EPS file.

When run from the command line it generates a random set of points inside a square of given length and finds the convex hull for those, printing the result as an EPS file.

Usage: `convexhull.py <numPoints> <squareLength> <outFile>`

Dinu C. Gherman

Small Bug: Only works with a list of UNIQUE points, Evan Jones, 2005/05/18 If the list of points passed to this function is not unique, it will raise an assertion. To fix this, remove these lines from the beginning of the `convexHull` function:

Taken from <http://aspn.activestate.com/ASPN/Cookbook/Python/Recipe/66527> and modified to work with complex numbers.

18.1 Functions

<code>saveAsEps($P, H, boxSize, path$)</code>

Save some points and their convex hull into an EPS file.
--

<code>convexHull(P)</code>
--

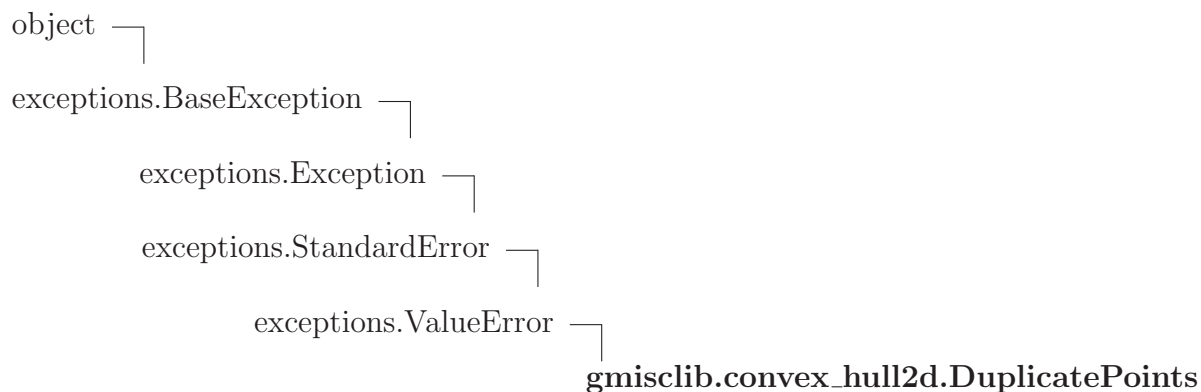
Calculate the convex hull of a set of complex points. If the hull has a duplicate point, an exception will be raised. It is up to the application not to provide duplicates.
--

<code>test()</code>

18.2 Variables

Name	Description
<code>epsHeader</code>	Value: <code>'%!PS-Adobe-2.0 EPSF-2.0\n%%%%BoundingBox: %d %d %d %d\n...'</code>
<code>__package__</code>	Value: <code>'gmisclib'</code>

18.3 Class *DuplicatePoints*



18.3.1 Methods

`__init__(self, s)`
`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature
 Overrides: `object.__init__` extit(inherited documentation)

Inherited from `exceptions.ValueError`

`__new__()`

Inherited from `exceptions.BaseException`

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

Inherited from `object`

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

18.3.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
<code>args</code> , <code>message</code>	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

19 Module `gmisclib.dict_vector`

Vectors of numbers, but indexed as a dictionary.

19.1 Functions

`round(x)`

`log(x)`

`min_within(x)`

`max_within(x)`

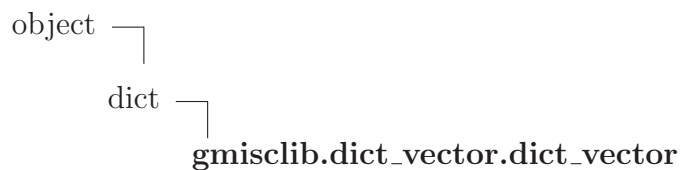
`min_between(* x)`

`max_between(* x)`

19.2 Variables

Name	Description
<code>--package--</code>	Value: <code>'gmisclib'</code>

19.3 Class `dict_vector`



19.3.1 Methods

`copy(self)`

Return Value

a shallow copy of D

Overrides: `dict.copy` `exitit`(inherited documentation)

incr(*self*, *k*, *v*)

__add__(*self*, *other*)

__gt__(*self*, *other*)

$x > y$

Overrides: `dict.__gt__` `exitit`(inherited documentation)

__lt__(*self*, *other*)

$x < y$

Overrides: `dict.__lt__` `exitit`(inherited documentation)

__ge__(*self*, *other*)

$x \geq y$

Overrides: `dict.__ge__` `exitit`(inherited documentation)

__le__(*self*, *other*)

$x \leq y$

Overrides: `dict.__le__` `exitit`(inherited documentation)

__mul__(*self*, *other*)

__div__(*self*, *other*)

__iadd__(*self*, *other*)

__isub__(*self*, *other*)

__sub__(*self*, *other*)

__radd__(*self*, *other*)

__rmul__(*self*, *other*)

__rsub__(*self*, *other*)

`--abs--(self)``--invert--(self)``sum(self)``median(self)``all(self)``float(self)``int(self)``iint(self)``ifloat(self)``sign(self)``round(self)``iround(self)``idropzero(self)``dropzero(self)`***Inherited from dict***

`--cmp--()`, `--contains--()`, `--delitem--()`, `--eq--()`, `--getattribute--()`, `--getitem--()`,
`--init--()`, `--iter--()`, `--len--()`, `--ne--()`, `--new--()`, `--repr--()`, `--setitem--()`, `--sizeof--()`,
`clear()`, `fromkeys()`, `get()`, `has_key()`, `items()`, `iteritems()`, `iterkeys()`, `itervalues()`,
`keys()`, `pop()`, `popitem()`, `setdefault()`, `update()`, `values()`

Inherited from object

`--delattr--()`, `--format--()`, `--reduce--()`, `--reduce_ex--()`, `--setattr--()`, `--str--()`, `--subclasshook--()`

19.3.2 Properties

Name	Description
<i>Inherited from object</i> __class__	

19.3.3 Class Variables

Name	Description
SCALARS	Value: (<type 'int'>, <type 'float'>)
ZERO	Value: 0
<i>Inherited from dict</i> __hash__	

20 Module *gmisclib.dictops*

Operations on dictionaries.

Version: \$Revision: 1.9 \$

20.1 Functions

filter(*d*, *l*)

Passes through items listed in *l*.

Parameters

l: list of keys to preserve

(*type=list*)

d: dictionary to copy/modify.

(*type=mapping*)

Return Value

a dictionary where all keys not listed in *l* are removed; all entries listed in *l* are copied into the (new) output dictionary.

(*type=dict*)

remove(*d*, *l*)

Removes items listed in *l*.

Parameters

l: list of keys to remove

(*type=list*)

d: dictionary to copy/modify.

(*type=mapping*)

Return Value

a dictionary where all keys listed in *l* are removed; all entries not listed in *l* are copied into the (new) output dictionary.

(*type=dict*)

rev1to1(*d*)

Reverse a 1-1 mapping so it maps values to keys instead of keys to values.

Parameters

d: (*type=dict*)

Return Value

M such that if $m[k]=v$, $M[v]=k$
(*type=dict*)

Raises

ValueError if the mapping is many-to-one.

compose(*d2*, *d1*)

Compose two mappings (dictionaries) into one.

Parameters

d2: second mapping

d1: first mapping

Return Value

d such that $d[k] = d2[d1[k]]$
(*type=dict*)

read2c(*f*)

Read in a dictionary from a two-column file; columns are separated by whitespace. This is not a completely general format, because keys cannot contain whitespace and the values cannot start or end in whitespace.

The format can contain comments at the top; these are lines that begin with a hash mark (#). Empty lines are ignored.

Parameters

f: an open file

Return Value

a dictionary and a list of comments
(*type=tuple(dict(str:str), list(str))*)

Note: You can get into trouble if you have a key that starts with a hash mark (#). If it happens to be the first key, and if there are no comments or if it directly follows a comment, it will be interpreted as a comment. To avoid this possibility, begin your file with a comment line, then a blank line.

argmax(*d*)

For a dictionary of key=value pairs, return the key with the largest value.

Raises

`ValueError` when the input dictionary is empty.

Note: When there are several equal values, this will return a single, arbitrary choice.

add_dol(*dict, key, value*)

OBSOLETE: replace with class `dict_of_lists`. Add an entry to a dictionary of lists. A new list is created if necessary; else the value is appended to an existing list. Obsolete: replace with `dict_of_lists`.

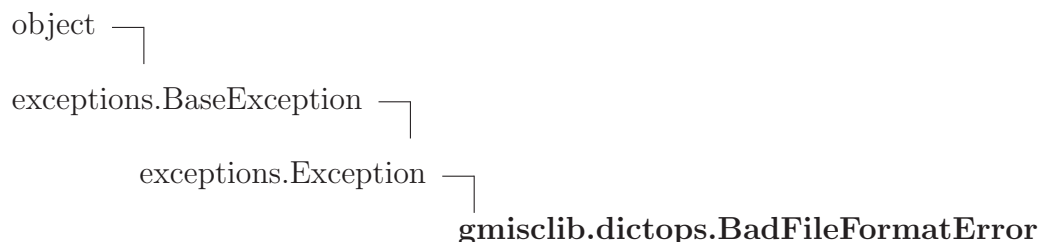
add_doc(*dict, key, value*)

OBSOLETE: replace with class `dict_of_accums`. Add an entry to a dictionary of counters. A new counter is created if necessary; else the value is added to an existing counter. Obsolete: replace with `dict_of_accums`.

20.2 Variables

Name	Description
<code>__package__</code>	Value: <code>None</code>

20.3 Class `BadFileFormatError`



This indicates a problem when reading in a dictionary via `read2c`.

20.3.1 Methods

`__init__(self, *s)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` extit(inherited documentation)

Inherited from `exceptions.Exception`

`__new__()`

Inherited from `exceptions.BaseException`

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

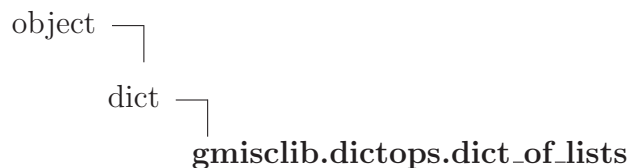
Inherited from `object`

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

20.3.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
<code>args</code> , <code>message</code>	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

20.4 Class dict_of_lists



A dictionary of lists.

20.4.1 Methods

`add(self, key, value)`

Append value to the list indexed by key.

addgroup (<i>self</i> , <i>key</i> , <i>values</i>)
--

Append values to the list indexed by key.

add_ifdifferent (<i>self</i> , <i>key</i> , <i>value</i>)
--

Append value to the list indexed by key.
--

copy (<i>self</i>)

This does a shallow copy.

Return Value

a shallow copy of D

Overrides: dict.copy

merge (<i>self</i> , <i>other</i>)

Inherited from dict

`__cmp__()`, `__contains__()`, `__delitem__()`, `__eq__()`, `__ge__()`, `__getattr__()`, `__getitem__()`, `__gt__()`, `__init__()`, `__iter__()`, `__le__()`, `__len__()`, `__lt__()`, `__ne__()`, `__new__()`, `__repr__()`, `__setitem__()`, `__sizeof__()`, `clear()`, `fromkeys()`, `get()`, `has_key()`, `items()`, `iteritems()`, `iterkeys()`, `itervalues()`, `keys()`, `pop()`, `popitem()`, `setdefault()`, `update()`, `values()`

Inherited from object

`__delattr__()`, `__format__()`, `__reduce__()`, `__reduce_ex__()`, `__setattr__()`, `__str__()`, `__subclasshook__()`

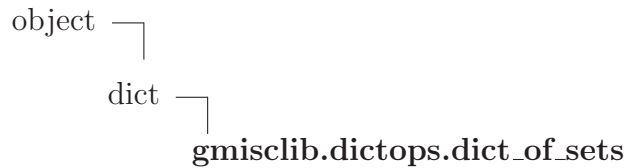
20.4.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

20.4.3 Class Variables

Name	Description
<i>Inherited from dict</i>	
<code>__hash__</code>	

20.5 Class `dict_of_sets`



A dictionary of lists.

20.5.1 Methods

<code>add(self, key, value)</code>

Append value to the set indexed by key.

<code>addgroup(self, key, values)</code>

Append values to the list indexed by key.

<code>copy(self)</code>

This does a shallow copy.

Return Value

a shallow copy of D

Overrides: <code>dict.copy</code>

<code>merge(self, other)</code>
--

Inherited from `dict`

`__cmp__()`, `__contains__()`, `__delitem__()`, `__eq__()`, `__ge__()`, `__getattr__()`, `__getitem__()`, `__gt__()`, `__init__()`, `__iter__()`, `__le__()`, `__len__()`, `__lt__()`, `__ne__()`, `__new__()`, `__repr__()`, `__setitem__()`, `__sizeof__()`, `clear()`, `fromkeys()`, `get()`, `has_key()`, `items()`, `iteritems()`, `iterkeys()`, `itervalues()`, `keys()`, `pop()`, `popitem()`, `setdefault()`, `update()`, `values()`

Inherited from `object`

`__delattr__()`, `__format__()`, `__reduce__()`, `__reduce_ex__()`, `__setattr__()`, `__str__()`, `__subclasshook__()`

20.5.2 Properties

Name	Description
<i>Inherited from object</i>	

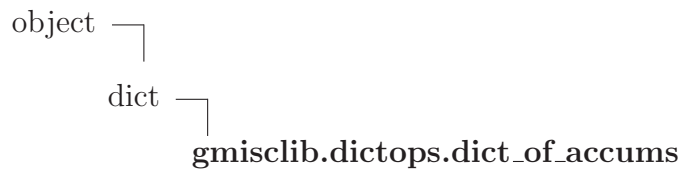
continued on next page

Name	Description
<code>--class--</code>	

20.5.3 Class Variables

Name	Description
<i>Inherited from dict</i>	
<code>--hash--</code>	

20.6 Class `dict_of_accums`



A dictionary of accumulators. Note that the `copy()` function produces a `dict_of_accums`.

20.6.1 Methods

`add(self, key, value)`

Add value to the value indexed by key.

`copy(self)`

Return Value

a shallow copy of D

Overrides: `dict.copy` `exitit`(inherited documentation)

`merge(self, other)`

Inherited from dict

`--cmp--()`, `--contains--()`, `--delitem--()`, `--eq--()`, `--ge--()`, `--getattribute--()`, `--getitem--()`, `--gt--()`, `--init--()`, `--iter--()`, `--le--()`, `--len--()`, `--lt--()`, `--ne--()`, `--new--()`, `--repr--()`, `--setitem--()`, `--sizeof--()`, `clear()`, `fromkeys()`, `get()`, `has_key()`, `items()`, `iteritems()`, `iterkeys()`, `itervalues()`, `keys()`, `pop()`, `popitem()`, `setdefault()`, `update()`, `values()`

Inherited from object

`--delattr--()`, `--format--()`, `--reduce--()`, `--reduce_ex--()`, `--setattr--()`, `--str--()`, `--subclasshook--()`

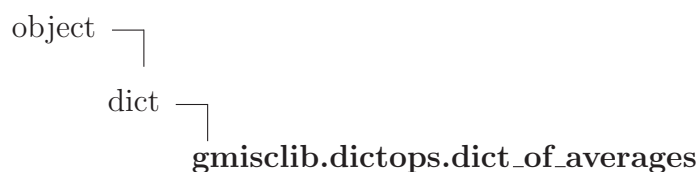
20.6.2 Properties

Name	Description
<i>Inherited from object</i> <code>__class__</code>	

20.6.3 Class Variables

Name	Description
<i>Inherited from dict</i> <code>__hash__</code>	

20.7 Class `dict_of_averages`



A dictionary of accumulators. Note that the `copy()` function produces a `dict_of_averages`.

20.7.1 Methods

<code>add(self, key, value, weight=1.0)</code> Add value to the value indexed by key.
<code>get_avg(self, key)</code>
<code>get_avgs(self)</code>
<code>get_both(self, key)</code>
<code>copy(self)</code> Return Value a shallow copy of D Overrides: <code>dict.copy</code> <code>extit</code> (inherited documentation)

<code>merge(self, other)</code>

Inherited from `dict`

`__cmp__()`, `__contains__()`, `__delitem__()`, `__eq__()`, `__ge__()`, `__getattr__()`, `__getitem__()`, `__gt__()`, `__init__()`, `__iter__()`, `__le__()`, `__len__()`, `__lt__()`, `__ne__()`, `__new__()`, `__repr__()`, `__setitem__()`, `__sizeof__()`, `clear()`, `fromkeys()`, `get()`, `has_key()`, `items()`, `iteritems()`, `iterkeys()`, `itervalues()`, `keys()`, `pop()`, `popitem()`, `setdefault()`, `update()`, `values()`

Inherited from `object`

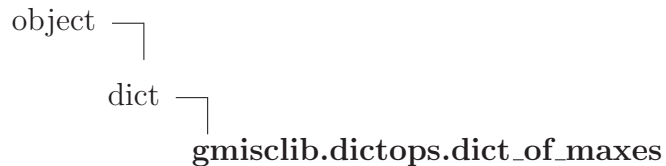
`__delattr__()`, `__format__()`, `__reduce__()`, `__reduce_ex__()`, `__setattr__()`, `__str__()`, `__subclasshook__()`

20.7.2 Properties

Name	Description
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

20.7.3 Class Variables

Name	Description
<i>Inherited from <code>dict</code></i>	
<code>__hash__</code>	

20.8 Class `dict_of_maxes`

A dictionary of maxima. Add a new number and the stored maximum will increase iff the new value is bigger. Note that the `copy()` function produces a `dict_of_accums`.

20.8.1 Methods

<code>add(self, key, value)</code>

Add value to the value indexed by key.
--

`copy(self)`

Return Value

a shallow copy of D

Overrides: `dict.copy` `exitit`(inherited documentation)

`merge(self, other)`

Inherited from dict

`__cmp__()`, `__contains__()`, `__delitem__()`, `__eq__()`, `__ge__()`, `__getattribute__()`, `__getitem__()`, `__gt__()`, `__init__()`, `__iter__()`, `__le__()`, `__len__()`, `__lt__()`, `__ne__()`, `__new__()`, `__repr__()`, `__setitem__()`, `__sizeof__()`, `clear()`, `fromkeys()`, `get()`, `has_key()`, `items()`, `iteritems()`, `iterkeys()`, `itervalues()`, `keys()`, `pop()`, `popitem()`, `setdefault()`, `update()`, `values()`

Inherited from object

`__delattr__()`, `__format__()`, `__reduce__()`, `__reduce_ex__()`, `__setattr__()`, `__str__()`, `__subclasshook__()`

20.8.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

20.8.3 Class Variables

Name	Description
<i>Inherited from dict</i>	
<code>__hash__</code>	

20.9 Class `dict_of_X`



A dictionary of arbitrary things. Note that the `copy()` function produces a `dict_of_X`.

20.9.1 Methods

`__init__(self, constructor, incremter)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Return Value

new empty dictionary

Overrides: `object.__init__` `exitit`(inherited documentation)

`add(self, key, *value)`

Add value to the thing indexed by key.

Inherited from dict

`__cmp__()`, `__contains__()`, `__delitem__()`, `__eq__()`, `__ge__()`, `__getattr__()`, `__getitem__()`, `__gt__()`, `__iter__()`, `__le__()`, `__len__()`, `__lt__()`, `__ne__()`, `__new__()`, `__repr__()`, `__setitem__()`, `__sizeof__()`, `clear()`, `copy()`, `fromkeys()`, `get()`, `has_key()`, `items()`, `iteritems()`, `iterkeys()`, `itervalues()`, `keys()`, `pop()`, `popitem()`, `setdefault()`, `update()`, `values()`

Inherited from object

`__delattr__()`, `__format__()`, `__reduce__()`, `__reduce_ex__()`, `__setattr__()`, `__str__()`, `__subclasshook__()`

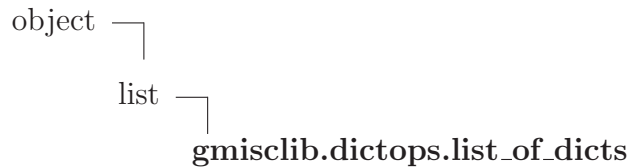
20.9.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

20.9.3 Class Variables

Name	Description
<i>Inherited from dict</i>	
<code>__hash__</code>	

20.10 Class `list_of_dicts`



20.10.1 Methods

`__init__(self, *arg)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Return Value

new empty list

Overrides: `object.__init__` extit(inherited documentation)

`lookup1(self, key, value)`

Inherited from `list`

`__add__()`, `__contains__()`, `__delitem__()`, `__delslice__()`, `__eq__()`, `__ge__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__gt__()`, `__iadd__()`, `__imul__()`, `__iter__()`, `__le__()`, `__len__()`, `__lt__()`, `__mul__()`, `__ne__()`, `__new__()`, `__repr__()`, `__reversed__()`, `__rmul__()`, `__setitem__()`, `__setslice__()`, `__sizeof__()`, `append()`, `count()`, `extend()`, `index()`, `insert()`, `pop()`, `remove()`, `reverse()`, `sort()`

Inherited from `object`

`__delattr__()`, `__format__()`, `__reduce__()`, `__reduce_ex__()`, `__setattr__()`, `__str__()`, `__subclasshook__()`

20.10.2 Properties

Name	Description
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

20.10.3 Class Variables

Name	Description
<i>Inherited from <code>list</code></i>	
<code>__hash__</code>	

21 Module *gmisclib.die*

21.1 Functions

die(*s*)

Output a fatal error message and terminate. Before the error message, it summarizes all the calls to **note**.

Parameters

s: error message

(*type=**str*)

warn(*s*)

Output a non-fatal warning. Before the warning, it summarizes all the calls to **note**.

Parameters

s: warning message

(*type=**str*)

info(*s*)

Output useful information. Before the message, it summarizes all the calls to **note**.

Parameters

s: message

(*type=**str*)

catch(*extext=***None**)

Call this inside an except statement. It will report the exception and any other information it has.

catchexit(*extext=***None**, *n=***1**, *text=***None**)

Call this inside an except statement. It will report all information and then exit.

dbg(*s*)

Output debugging information, if debug is nonzero. Before the debugging info, it prints all information given to `note`.

Parameters

s: debug message
(*type=*`str`)

exit(*n*, *text=None*)

Exit, after dumping accumulated `notes`.

Parameters

n: the processes' exit code
(*type=*`int`)
text: something to print
(*type=*`str` or `None`)

note(*key*, *value*)

Memorize a note, which will be output along with the next error/warning/info message. These will be printed in the form '`#NOTE key = value`', with value converted to a string at the time of printing. Note that only the most recent note is kept for each key.

Parameters

value: (*type=*`anything`)
key: (*type=*`str`)

get(*key*)

Get the most recent value memorized by a call to `note`.

Parameters

key: (*type=*`str`)

Return Value

whatever was memorized

21.2 Variables

Name	Description
debug	Set to False to disable <code>dbg()</code> messages. Value: True

continued on next page

Name	Description
nocompress	False means that only the first of repeated messages are displayed; True displays all. Value: False
Threads	Value: False
--package--	Value: 'gmisclib'

21.3 Class counter

21.3.1 Methods

```
__init__(self)
```

```
set_out_err(self, stdout, stderr)
```

```
incoming(self, msgtype, name, s)
```

```
showreps(self)
```

```
showdump(self, s)
```

```
__del__(self)
```

```
clearmem(self)
```

```
dumpmem(self)
```

```
memorize(self, key, value)
```

21.3.2 Class Variables

Name	Description
lock	Value: threading.Lock()

22 Module `gmisclib.die2`

22.1 Functions

note (<i>name</i> , <i>val</i>)
--

show (<i>name</i>)

die (<i>s</i>)

Output a fatal error message and terminate.

warn (<i>s</i>)

Output a non-fatal warning.

info (<i>s</i>)

Output useful information.

catch (<i>extext</i> =None)

Call this inside an except statement. It will report the exception and any other information it has.
--

catchexit (<i>extext</i> =None, <i>n</i> =1, <i>text</i> =None)

Call this inside an except statement. It will report all information and then exit.

dbg (<i>s</i>)

Output debugging information, if debug is nonzero.
--

exit (<i>n</i> , <i>text</i> =None)

Exit, after dumping accumulated messages.

get (<i>key</i>)

22.2 Variables

Name	Description
debug	Value: True

continued on next page

Name	Description
compress	Value: True
logfd	Value: sys.stderr
__package__	Value: 'gmisclib'

23 Module gmisclib.do_exec

Execute a *nix command and capture the output.

Version: \$Revision: 1.5 \$

23.1 Functions

getall(*s*)

Read a list of lines from a process, after dropping junk like comments (beginning with whitespace then '#') or blank lines. Raises an exception if if the process returns a line beginning with 'ERR:' The argument *s* is a string containing the command and its arguments. *S* is normally passed to a shell.

get(*s*)

Read a single line from a process, after dropping junk like comments (#) or blank lines. Returns a trouble indication if the process returns 'ERR:'

23.2 Variables

Name	Description
--package--	Value: 'gmisclib'

24 Module *gmisclib.do_pgplot*

Version: \$Revision: 1.1 \$

24.1 Variables

Name	Description
<code>pl</code>	Value: <code>pgplot.cpgplot</code>

24.2 Class *plot*

24.2.1 Methods

```
__init__(self, dev=None)
```

```
__del__(self)
```

```
next(self)
```

```
ask(self, onoff=1)
```

```
size(self, width, aspect)
```

```
panels(self, w, h)
```

```
flush(self)
```

```
start(self)
```

25 Module *gmisclib.ds9_region*

25.1 Variables

Name	Description
<code>--package--</code>	Value: <code>'gmisclib'</code>

25.2 Class writer

25.2.1 Methods

```
--init--(self, fd)
```

```
header(self, d)
```

```
text(self, x, y, text)
```

```
close(self)
```

```
--del--(self)
```

26 Module *gmisclib.dyn_prog*

Viterbi search

26.1 Functions

<code>path(<i>nodecost</i>, <i>linkcost</i>)</code>

<code>test1()</code>

<code>test2()</code>

<code>test3()</code>

26.2 Variables

Name	Description
<code>--package--</code>	Value: 'gmisclib'

27 Module `gmisclib.edit_distance`

Levenshtein (edit) distance between two strings of symbols.

27.1 Functions

dist(*s*, *t*, *csub*, *cinsert*, *cdel*)

Cost for converting string *s* into string *t*. This function takes three dictionaries that contain the cost of insertion, deletion, and substitutions.

Parameters

s: starting string
(*type=string or array*)

t: ending string
(*type=string or array*)

csub: cost of converting a to b
(*type=dict((a,b) : float)*)

cinsert: cost of insertion
(*type=dict(a: float)*)

cdel: cost of deletion
(*type=dict(a: float)*)

distf1(*s*, *t*, *csub*)

Cost for converting string *s* into string *t*. This function takes a function that computes the cost of insertions, deletions, or substitutions. (This is an alternative implementation for `distf2()`.)

Parameters

s: starting string
(*type=string or array*)

t: ending string
(*type=string or array*)

csub: cost of converting a to b
(*type=function(a,b) : float*)

distf2(*s*, *t*, *csub*)

Cost for converting string *s* into string *t*. This function takes a function that computes the cost of insertions, deletions, or substitutions.

Parameters

- s:** starting string
(*type=string or array*)
- t:** ending string
(*type=string or array*)
- csub:** cost of converting *a* to *b*
(*type=function(a,b) : float*)

distf(*s*, *t*, *csub*)

Cost for converting string *s* into string *t*. This function takes a function that computes the cost of insertions, deletions, or substitutions.

Parameters

- s:** starting string
(*type=string or array*)
- t:** ending string
(*type=string or array*)
- csub:** cost of converting *a* to *b*
(*type=function(a,b) : float*)

def_cost(*a*, *b*)

This is the default cost function for converting *a* into *b*. Insertions, deletions, and substitutions all have a unit cost. Normally, this is passed as an arg to `distf`.

Parameters

- a:** a symbol (or None to indicate an insertion).
- b:** a symbol (or None to indicate an deletion).

Return Value

- the cost of changing *a* into *b*
(*type=int*)

free_sub(*a*, *b*)

This is a sample cost function for converting **a** into **b**. Insertions and deletions have a unit cost; substitutions are free. Normally, this is passed as an arg to `distf`.

Parameters

a: a symbol (or `None` to indicate an insertion).

b: a symbol (or `None` to indicate an deletion).

Return Value

the cost of changing **a** into **b**

(*type=int*)

free_del(*a*, *b*)

This is a sample cost function for converting **a** into **b**. Insertions and substitutions have a unit cost; deletion are free. Normally, this is passed as an arg to `distf`.

Parameters

a: a symbol (or `None` to indicate an insertion).

b: a symbol (or `None` to indicate an deletion).

Return Value

the cost of changing **a** into **b**

(*type=int*)

test()**test2()**

27.2 Variables

Name	Description
<code>--package--</code>	Value: <code>'gmisclib'</code>

27.3 Class `text_cost`

object └─ `gmisclib.edit_distance.text_cost`

This is useful for differencing documents that have been parsed into lists of words. It computes the cost of a change of words by an edit distance computation on the characters within the words. (It caches the costs of commonly encountered words for speed.) Instances of this class can be passed as a cost function to `distf`.

27.3.1 Methods

`__init__(self, costfac=None, cachesize=None, word_cost=None, frac_exp=0.0)`

Create an instance.

Parameters

- costfac:** None, or an arbitrary function that multiplies the scaled edit distance between the two words. Called as `self.costfac(a,b)` where either `a` or `b` can be None for insertions or deletions.
- (type=function(str,str): float or None)*
- cachesize:** how large a cache of word-to-word distances should be kept?
- (type=None (meaning unlimited) or int)*
- word_cost:** cost function to use inside words: this is a cost for insertion, deletion, or substitution of letters.
- (type=function(str,str): float or None (meaning use def_cost).)*
- frac_exp:** An exponent for scaling the edit distance by the length of words.
- (type=float)*

Overrides: `object.__init__`

`__call__(self, a, b)`

To be used by `distf`.

Parameters

- a:** *(type=str or None)*
- b:** *(type=str or None)*

Return Value

- a cost
- (type=presumably a float)*
-

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

27.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

28 Module `gmisclib.entropy`

28.1 Functions

$P(x)$

<code>multinomial_logp(x, cF)</code>

<code>multinomial_fixer(x, c)</code>

<code>information_gained(c, pc=None)</code>

Suppose there is an experiment where you have N conditions and in each conditions there are M possible measurements. The outcomes are represented by `c[N,M]` matrix of counts: how many times you observe each possible measurement for each condition.

We want to ask what's the information gained by taking a measurement. We assume the probability of each condition is a `pc[N]` vector, which defaults to the frequencies in `c`. (We assume that the choice of condition is not a random variable, but is made in advance, like most experiments.)

28.2 Variables

Name	Description
<code>--package--</code>	Value: <code>'gmisclib'</code>

29 Module *gmisclib.erb_scale*

ERB Perceptual frequency scale.

Version: \$Revision: 1.6 \$

29.1 Functions

f_to_erb (<i>f</i>)
Frequency to ERB band number. http://ccrma-www.stanford.edu/~jos/bbt/Equivalent_Rectangular_Bandwidth.html

erb_to_f (<i>e</i>)
ERB number -> frequency

ebw (<i>e</i>)
Critical bandwidth (in Hz) at frequency <i>e</i> (erbs).

29.2 Variables

Name	Description
<code>--package--</code>	Value: <code>'gmisclib'</code>

30 Module *gmisclib.evolution*

30.1 Variables

Name	Description
<code>--package--</code>	Value: None

31 Module gmisclib.fake_file

A class that implements a simple file in memory. It's not an exact simulation of a real file: it assumes that no seeking is done, and that all reads are line-at-a-time. It also (intentionally) allow reading from files opened in mode 'r', and allows writing for files opened in mode 'w'.

Version: \$Revision: 1.4 \$

31.1 Functions

open (<i>name</i> , <i>mode</i>)

31.2 Variables

Name	Description
<code>--package--</code>	Value: 'gmisclib'

31.3 Class file

StringIO.StringIO —
gmisclib.fake_file.file

31.3.1 Methods

__init__ (<i>self</i> , <i>mode</i> ='rw', <i>name</i> =None) Overrides: StringIO.StringIO.__init__
--

Inherited from StringIO.StringIO

`__iter__()`, `close()`, `flush()`, `getvalue()`, `isatty()`, `next()`, `read()`, `readline()`, `readlines()`, `seek()`, `tell()`, `truncate()`, `write()`, `writelines()`

32 Module *gmisclib.fiat_merge*

32.1 Functions

`merge(d1, d2, key1, key2, grab, unique=True)`

32.2 Variables

Name	Description
DEBUG	Value: 0
__package__	Value: 'gmisclib'

33 Module *gmisclib.fiatio*

Fiatio reads and writes an extension of the FIAT file format originally defined by David Wittman (UC Davis). This reads and writes the FIAT 1.2 format, defined at <http://kochanski.org/gpk/paper> (FIAT 1.0 is defined at <http://dls.physics.ucdavis.edu/fiat/fiat.html>.)

Nice FIAT features:

- header information looks like a comment to most programs, so they will treat a FIAT file as simple multi-column ASCII.
- since it has column names in the header, you can add columns at will, and your existing scripts will continue to run.
- Simple to parse.
- Easy to generate.

This describes fiat 1.2 format, which is nearly 100% upwards compatible with fiat 1.0 format. It is defined as follows:

1. Lines are separated by newlines.
2. All values are encoded by replacing newline and other difficult characters by a percent character (%) followed by a hex code on writing, and the reverse on reading. (There are also some more human-friendly codes which can be used, instead of pure hex: see `g_encode_specials` for their definitions. Notably, %S is space, %L is newline, %R is carriage return, %t is tab, and %T is percent.)
3. At the top of the file, you have a line identifying the format: "# fiat 1.2" (regexp: "# fiat 1\.[0-9.]+").
4. Then, you typically have a number of header lines beginning with "#". Header lines are in the form `# attribute = value` (where white space is optional and can be a mixture of spaces and tabs). The attribute must match the regular expression `[a-zA-Z][a-zA-Z_0-9]*`. The value is whatever follows the equals sign, after leading and following white space is stripped. If the value begins and ends with the same quote character, either ' or ", the quotes are also stripped off. Values may contain any character except newline and the chosen quote.
 - Note that you must quote or encode a value if it begins or ends with whitespace.
 - Note also that header lines can also appear further down in the file; they are not restricted to the top.
 - Any other header lines are just treated as comments and ignored.
1. There may be header lines of the form "# TTYPE1 = name" or "#ttype4 = name" which name the columns of the data (the leftmost column is TTYPE1). If you don't name the Ith column, its name will be I. When writing, this module adds an attribute `COL_SEPARATOR` which contains the numeric code(s) (ASCII) of the column separator character(s). This defaults to 9, ASCII tab. The module also adds a `COL_EMPTY` attribute with the string used to mark an blank (nonexistent) item. (This defaults to

`%na`.) Note that nonexistent is not the same as a zero-length string.

- These lines may also appear anywhere in the file. They take effect immediately.
- All attributes and names are optional.

- Typically, the header is followed by multicolumn ASCII data. Columns are separated (by default) with any white space, but if there is a `COL_SEPARATOR` attribute, it is used instead. Empty entries for columns should be indicated by whatever code is specified in `COL_EMPTY`, if that is set. Otherwise, if `COL_SEPARATOR` is set, `COL_SEPARATOR` strings separate items, some of which may simply be empty. (In all cases, a completely blank line is treated as a datum which has all columns blank (nonexistent).)

If there is no `DATE` attribute, the write routine adds one, using the current date and time, in the form `ccyy-mm-ddThh:mm:ss` (as defined in the NASA FITS format). Note that all attributes are optional.

This is not quite David Wittman FIAT (1.0), which forces value to either be quoted or to contain no white space. Dwittman FIAT will take a line in the form `"#a=b c"`, and interpret `c` as a comment, whereas fiat 1.2 will interpret the value as `"b c"`. However, almost all files will be interpreted the same way as Fiat 1.0.

Here's an example:

```
# fiat 1.2
# TTYPE1 = b
# TTYPE2 = a
# SAMPRATE = 2.3
# DATE = 2001-09-21T21:32:32
# COL_EMPTY = "%na"
# COL_SEPARATOR = "9"
# Comment1
# Comment2
# b      a
2        1
3        2
3        %na
%na      3
%na      %na
0        1
```

33.1 Functions

<code>col_order(a, b)</code>

```
write_array(fd, adata, columns=None, comments=None, hdr=None,  
sep='\t', numeric=False)
```

Write a rectangular array in FIAT format. Adata is a 2-D numpy array or a sequence of sequences.

```
write(fd, ldata, comments=None, hdr=None, sep='\t', blank='%na',  
fixed_order=0)
```

Write a file in FIAT format. Ldata is a list of dictionaries. Each dictionary corresponds to one line in the file. Each unique key generates a column, and the values are printed in the data section of the FIAT file. Note that the TTYPE header lines will be automatically generated from ldata. Hdr is a dictionary of information that will be put in the header. Comments is a list of comment lines for the header. Fd is a file descriptor where the data should be written. Sep is a string used to separate data columns. Blank is a string to use when a data value is missing.

```
shared_data_values(data_items)
```

Takes a list of data and pulls out all the items that have the same value in each line. The idea is that you can then put them into the header via:

```
hdr, data, c = read(fd) htmp, data = shared_data_values(data)
hdr.update(htmp)
```

Parameters

data_items: (*type=list(dict(str: anything))*)

Return Value

It returns a tuple of (1) a dictionary of header items, and (2) a list of data. The list has the same length as *data_items*, but the dictionaries within it may have fewer entries.

(*type=tuple(dict(str: anything),*
list(dict(str: anything))))

read(*fd*)

Read a fiat format data file. Each line in the FIAT file is represented by a dictionary that maps column name into the data (data is a string). Lines without data in a certain column will not have the corresponding entry in the dictionary for that line.

You can use this function as follows:

```
hdr, data, comments = read(fd)
for datum in data:
    print datum['X']
```

Parameters

fd: The data source: typically a file descriptor.

*(type=An iterator that generates strings. Typically a **file** object.)*

Return Value

Three items: header, data, and comments. Header is the collected dictionary of header information data is a list of dictionaries, one for each line in the file, and comments is a list of strings.

(type=tuple(dict, list(dict(str:str)), list(str)))

read_merged(*fd*)

Read in a fiat file and return a list of dictionaries, one for each line in the file. (Also a list of comment lines.) Each line in the input FIAT file is represented by a dictionary that maps column name into the data (data is a string). The header data in the FIAT file is merged into the per-column data, so that the header data is used as a default value for the column of the same name. As a result, all the information in the file (both header and data) is in the resulting list of dictionaries.

NB: this is a bit of a specialized routine. Normally, one uses **read**.

E.g. if there is a header line "`# X = Y`" and no data column called "`X`", then this will succeed:

```
data, comments = read_merged(fd):
for datum in data:
    assert datum['X']=='Y'
```

That this routine does not require header lines to precede data lines. If header lines appear in the middle, then a new column will be created from that point onwards.

Parameters

fd: The data source: typically a file descriptor.

*(type=An iterator that generates strings. Typically a **file** object.)*

Return Value

(data, comments)

(type=(list(dict), list(str)))

readiter(*fd*)

Read in a fiat file. Each line in the file is represented by a dictionary that maps column name into the data (data is a string). Lines without data in a certain column will not have an entry in that line's dictionary for that column name.

Lines beginning with '#' are either header or comment lines. A fiat file can mix header lines amongst the data lines. (Although, typically, all the header info is at the top.)

You can use this function as follows:

```
for (hdr, datum, comments) in readiter(fd):
    print datum['X']
```

NB: this is a bit of a specialized routine. Normally, one uses `read`.

Parameters

fd: The data source: typically a *file*. Not a filename.

(type=Anything that supports iteration. Typically a file object.)

Return Value

Three items: header, data, and comments. Header is the collected dictionary of header information since the last iteration, data is a dictionary of the data on the current line, and comments is a list of comment string seen so far. The end of the file yields None for the last data, along with any header info or comments after the last datum.

(type=(dict(str:str), dict(str:str), list(str)))

read_as_float_array(*fd*, loose=False, baddata=None)

Read in a fiat file. Return (header, data, comments), where header is a dictionary of header information, data is a numpy array, and comments is a list of strings. Two special entries are added to header: `__COLUMNS` points to a mapping from column numbers (the order in which they appeared in the file, left to right, starting with 0) to names, and `__NAME_TO_COL` holds the reverse mapping.

Empty values are set to NaN.

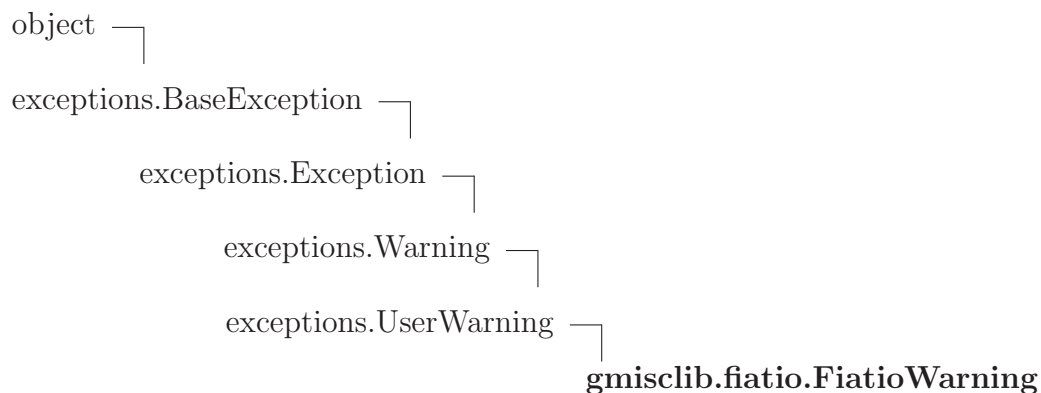
If `loose==False`, then all entries must be either floating point numbers or empty entries or equal to `baddata` (as a string, before conversion to a float). If `loose==True`, all non-floats will simply be masked out.

`test1()``test2()``test3()``test4()``test()`

33.2 Variables

Name	Description
<code>TABLEN</code>	Value: 8
<code>__package__</code>	Value: 'gmisclib'

33.3 Class *FiatioWarning*



33.3.1 Methods

```

__init__(self, *s)
x.__init__(...) initializes x; see x.__class__.__doc__ for signature
Overrides: object.__init__ extit(inherited documentation)

```

Inherited from exceptions.UserWarning

`__new__()`

Inherited from exceptions.BaseException

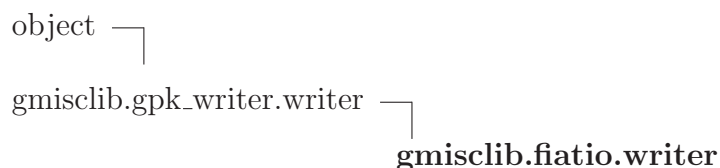
`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

Inherited from object

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

33.3.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
<code>args</code> , <code>message</code>	
<i>Inherited from object</i>	
<code>__class__</code>	

33.4 Class writer

Known Subclasses: `gmisclib.fatio.merged_writer`

Write a file in FIAT format. This class represents an open file, and you call member functions to write data into the file. This automatically generates much of the header information.

Column names are set from the keys passed in the `datum()` method. Each unique key generates a column, and the values are printed in the data section of the FIAT file. The TTYPE header lines will also be automatically generated.

33.4.1 Methods

<code>comment(self, comment)</code>
Add a comment to the data file.
Parameters
<code>comment</code> : the comment
<i>(type=str)</i>
Overrides: <code>gmisclib.gpk_writer.writer.comment</code>

header(*self*, *k*, *v*)

Add a single **key=value** line to the header of the data file.

Parameters

k: key
 (*type=**str*)

v: value
 (*type=**str*)

Overrides: gmisclib.gpk_writer.writer.header

__init__(*self*, *fd*, *sep*='\\t', *blank*='%na')

x.__init__(...) initializes *x*; see *x.__class__.__doc__* for signature

Parameters

fd: where to write the data
 (*type=**file*)

sep: what separates columns?
 (*type=**str*)

blank: what marks a spot where there isn't data?
 (*type=**str*)

Overrides: object.__init__

add_cols(*self*, *colnames*)

datum(*self*, *data_item*)

Write a line into a fiat file. They column names will be set from the keys.

Parameters

data_item: a dictionary of **key=value** pairs.
 (*type=**dict*(*str*: *anything*))

Overrides: gmisclib.gpk_writer.writer.datum

datavec(*self*, *vector*, *numeric*=False)

This assumes that you've already called **add_cols()** to set the column names. It is an error to have a vector whose length doesn't match the number of column names.

append(*self*, *d*)

close (<i>self</i>)

comments (<i>self</i> , <i>comments</i>)

Add comments to the data file. Comments can appear anywhere.
--

data (<i>self</i> , <i>dataset</i>)
--

Write a series of lines to the output file.

extend (<i>self</i> , <i>d</i>)
--

flush (<i>self</i>)

headers (<i>self</i> , <i>h</i>)

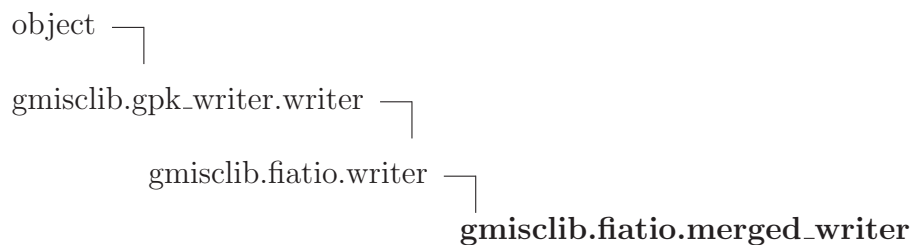
Inherited from object

`--delattr--()`, `--format--()`, `--getattr--()`, `--hash--()`, `--new--()`, `--reduce--()`, `--reduce_ex--()`,
`--repr--()`, `--setattr--()`, `--sizeof--()`, `--str--()`, `--subclasshook--()`

33.4.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

33.5 Class `merged_writer`



Assumes that the data will be read with `read_merged()`, so that header values will supply default values for each column.

33.5.1 Methods

`__init__(self, fd, sep='\t', blank='%na')`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Parameters

`fd`: where to write the data

`sep`: what separates columns?

`blank`: what marks a spot where there isn't data?

Overrides: `object.__init__` `exitit`(inherited documentation)

`header(self, k, v)`

Add a single **key=value** line to the header of the data file.

Parameters

`k`: key

`v`: value

Overrides: `gmisclib.gpk_writer.writer.header`

`datum(self, data_item)`

Assumes that the data will be read with `read_merged()`, so that header values will supply default values for each column. This writes a line in the fiat file, but first it deletes any values that already exist as a header item of the same name.

Parameters

`data_item`: a dictionary of **key=value** pairs.

Overrides: `gmisclib.gpk_writer.writer.datum`

`add_cols(self, colnames)`

`append(self, d)`

`close(self)`

comment(*self*, *comment*)

Add a comment to the data file.

Parameters

comment: the comment
(*type=**str*)

Overrides: *gmisclib.gpk_writer.writer.comment*

comments(*self*, *comments*)

Add comments to the data file. Comments can appear anywhere.

data(*self*, *dataset*)

Write a series of lines to the output file.

datavec(*self*, *vector*, *numeric=False*)

This assumes that you've already called *add_cols()* to set the column names. It is an error to have a vector whose length doesn't match the number of column names.

extend(*self*, *d*)

flush(*self*)

headers(*self*, *h*)

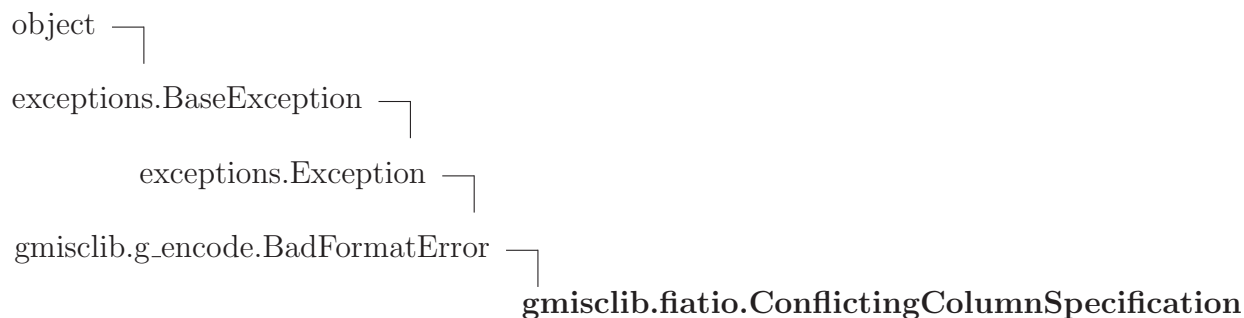
Inherited from object

__delattr__(), *__format__()*, *__getattr__()*, *__hash__()*, *__new__()*, *__reduce__()*, *__reduce_ex__()*, *__repr__()*, *__setattr__()*, *__sizeof__()*, *__str__()*, *__subclasshook__()*

33.5.2 Properties

Name	Description
<i>Inherited from object</i>	
<i>__class__</i>	

33.6 Class *ConflictingColumnSpecification*



33.6.1 Methods

`__init__(self, s)`
`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature
 Overrides: `object.__init__` extit(inherited documentation)

Inherited from exceptions.Exception

`__new__()`

Inherited from exceptions.BaseException

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

Inherited from object

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

33.6.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
	args, message
<i>Inherited from object</i>	
<code>__class__</code>	

34 Module *gmisclib.find_home*

This lets you find files with just a path name relative to where the program's executable code script sits.

Version: \$Revision: 1.6 \$

34.1 Functions

executable (<i>s</i> , <i>general</i> =True)
--

Find an executable program. It searches first in the directory of the currently executing python script. Optionally, it then looks at PATH. If <i>general</i> =False, the function should fail unless the executable is in the same directory as the currently executing python script.

module (<i>s</i> , <i>general</i> =True)
--

Find a module. Returns pathname.

os_prgm (<i>nm</i> , <i>general</i> =1)

Find an operating-system dependent executable program.
--

data (<i>s</i>)

Find a data file.

directory (<i>s</i>)

Find a directory.

34.2 Variables

Name	Description
<code>--package--</code>	Value: 'gmisclib'

35 Module *gmisclib.find_ngram*

This module lets you search through label files to find particular ngrams.

Version: \$Revision: 1.7 \$

35.1 Functions

matches (<i>data</i> , <i>pattern</i>)

find_mark (<i>ngram</i> , <i>fname</i> , <i>datatype</i> =1)
--

ngram = list of labels. *fname* = file name for *xwaves_mark* datafile. *datatype* = *xwaves_mark*.PHONE or *xwaves_mark*.WORD

This function returns all instances (even overlapping instances) of the specified N-gram in the file. The return format is [(end_time, label), ...], where the zeroth entry in the list is the symbol before the start of the N-gram. It's end time is the beginning of the N-gram.

In the argument list, the N-gram is an array of labels; the labels need to match the file's labels exactly.

find_lab (<i>ngram</i> , <i>fname</i> , <i>loose</i> =0)
--

ngram = list of labels. *fname* = file name for *xwaves_lab* datafile.

This function returns all instances (even overlapping instances) of the specified N-gram in the file. The return format is [(end_time, label), ...], where the zeroth entry in the list is the symbol before the start of the N-gram. It's end time is the beginning of the N-gram.

In the argument list, the N-gram is an array of labels; the labels need to match the file's labels exactly.

find(*ngram*, *data*)

ngram = list of labels. *data* = list of (time, label, ...) as produced by `xwaves_lab.py` or similar.

This function returns all instances (even overlapping instances) of the specified N-gram in the file. The return format is [[label, ...], ...]. It is a list of n-grams, and each n-gram is a list of entities, and each entities is a tuple which marks when it ends, what it is specifically, and perhaps other things.

In the argument list, the N-gram is an array of labels; the labels need to match the file's labels exactly.

35.2 Variables

Name	Description
<code>WildAnyOne</code>	Value: <code>re.compile(r'.')</code>
<code>--package--</code>	Value: <code>'gmisclib'</code>

36 Module gmisclib.findleak

Author: Samual M. Rushing

36.1 Functions

```
get_refcounts()
```

```
print_top_N( $N=100$ ,  $prefix=\text{'#T0Pref'}$ )
```

```
readvm()
```

```
alloc1k()
```

```
vm( $comment=\text{' '}$ )
```

```
lookvmstat( $fname$ )
```

36.2 Variables

Name	Description
--package--	Value: 'gmisclib'

37 Module *gmisclib.ftest*

37.1 Functions

fprob(*dfnum*, *dfden*, *F*)

Returns the (1-tailed) significance level (p-value) of an F statistic given the degrees of freedom for the numerator (*dfR*-*dfF*) and the degrees of freedom for the denominator (*dfF*).

Usage: *lfprob*(*dfnum*, *dfden*, *F*) where usually *dfnum*=*dfbn*, *dfden*=*dfwn*

betai(*a*, *b*, *x*)

Returns the incomplete beta function:

$I\text{-sub-}x(a,b) = 1/B(a,b) * (\text{Integral}(0,x) \text{ of } t^{(a-1)}(1-t)^{(b-1)} dt)$

where $a,b > 0$ and $B(a,b) = G(a) * G(b) / (G(a+b))$ where $G(a)$ is the gamma function of a . The continued fraction formulation is implemented here, using the *betacf* function. (Adapted from: Numerical Recipies in C.)

Usage: *lbetai*(*a*,*b*,*x*)

betacf(*a*, *b*, *x*)

This function evaluates the continued fraction form of the incomplete Beta function, *betai*. (Adapted from: Numerical Recipies in C.)

Usage: *lbetacf*(*a*,*b*,*x*)

gammln(*xx*)

Returns the gamma function of *xx*. $\Gamma(z) = \text{Integral}(0,\text{infinity}) \text{ of } t^{(z-1)} \exp(-t) dt$. (Adapted from: Numerical Recipies in C.)

Parameters

xx: float

Return Value

float

agammln(*xx*)

Returns the gamma function of *xx*. $\text{Gamma}(z) = \text{Integral}(0, \text{infinity}) \text{ of } t^{(z-1)} \exp(-t) dt$. Adapted from: Numerical Recipies in C. Can handle multiple dims ... but probably doesn't normally have to.

Usage: `agammln(xx)`

37.2 Variables

Name	Description
--package--	Value: 'gmisc.lib'

38 Module gmisclib.fuzzygraph

38.1 Functions

<code>addcurve(<i>xyee</i>, <i>x_range</i>, <i>y_range</i>, <i>bexp</i>, <i>brms</i>, <i>imopfd</i>, <i>xsz</i>, <i>ysz</i>)</code>

38.2 Variables

Name	Description
IB	Value: 0.7
__package__	Value: 'gmisclib'

39 Module `gmisclib.g2_select`

39.1 Functions

filterlist(*s*, *d*, *verbose*=False)

This filters a list of dictionaries, passing ones where the *selector* returns true.

Parameters

s: the selector – a little snippet of python

(*type*=*str*)

d: the list of dictionaries. Each dictionary is temporarily used as the local variable space and the selector is evaluated.

(*type*= *list(dict(str:value))*)

Return Value

a subset of the input list, whichever dictionaries cause **s** to return True when evaluated.

(*type*= *list(dict(str: value))*)

filter_iter(*s*, *d*, *verbose*=False)

This filters a list of lines, passing ones where the selector returns true. *S* is the selector – a little snippet of python, and *d* is the list of data : a list of {k:v} mappings.

accept(*s*, *d*)

This checks a single dictionary, and returns the result of the selector. Errors in the evaluation are trapped and cause `accept()` to return False. *S* is the selector – a little snippet of python, and *d* is the data : a {k:v} mapping.

whynot(*s*, *d*)

Returns an explanation of why *d* was not accepted, given *s*, or None if *d* was accepted.

why(*s*, *d*)

Returns an explanation of why *d* was accepted, given *s*, or None if *d* was not accepted.

evaluate(*s*, *d*)

This checks a single dictionary, and returns the result of the selector. S is the selector – a little snippet of python, and d is the data : a {k:v} mapping.

test()

39.2 Variables

Name	Description
CCSZ	Value: 100
__package__	Value: 'gmisclib'

39.3 Class selector_c

object └─ gmisclib.g2_select.selector_c

39.3.1 Methods

__init__(*self*, *code*, *global_values*=None)

Code = a string containing python code. Global_values = a dictionary containing values to be used by that python code. Note that the dictionary is not copied, so that it is shared and changes will be noticed.

Overrides: object.__init__

set_code(*self*, *code*)

set_trap(*self*, *exc*, *result*)

This adds to a list of exceptions. If any of those exceptions happen later, when you run **self.eval()**, then they will be caught and turned into the specified result.

For instance, if you call **set_trap(ValueError, False)**, and when you execute **self.cc** in **self.eval** a **ValueError** is raised, then **self.eval()** will return **False**. You have mapped a **ValueError** exception into a returned value of **False**.

<code>eval(self, locals)</code>

<code>globals(self, s)</code>

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

39.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

40 Module gmisclib.g_closure

This module defines closures. Closures are functions that have been supplied with some arguments, but not all of them.

40.1 Variables

Name	Description
<code>__package__</code>	Value: None

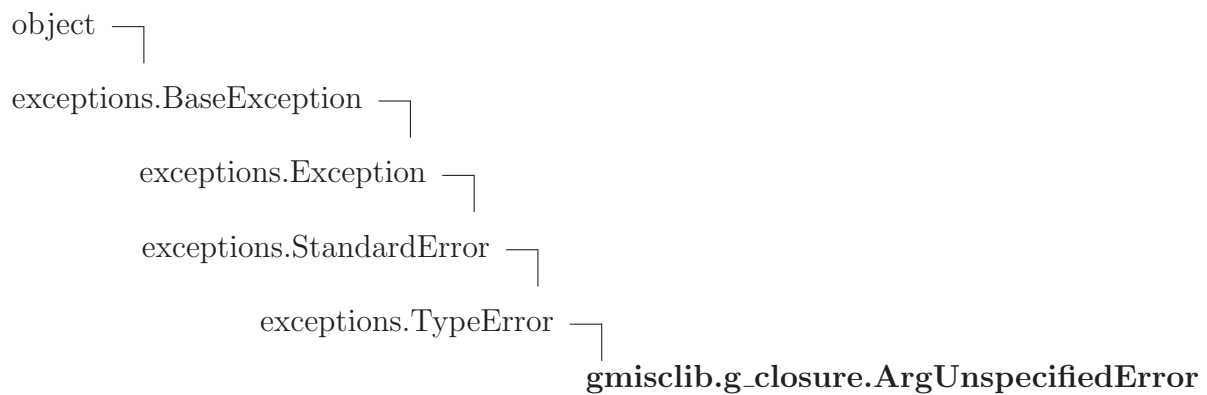
40.2 Class NotYet

Just a marker, indicating that a certain argument is not yet present, and the actual call should be deferred until later.

40.2.1 Methods

<code>__repr__(self)</code>

40.3 Class ArgUnspecifiedError



40.3.1 Methods

`--init--`(*self*, *s*)

x.`--init--`(...) initializes *x*; see *x*.`--class--`.`--doc--` for signature

Overrides: `object.--init--` `exitit`(inherited documentation)

Inherited from exceptions.TypeError

`--new--`()

Inherited from exceptions.BaseException

`--delattr--`(), `--getattr--`(), `--getitem--`(), `--getslice--`(), `--reduce--`(), `--repr--`(),
`--setattr--`(), `--setstate--`(), `--str--`(), `--unicode--`()

Inherited from object

`--format--`(), `--hash--`(), `--reduce_ex--`(), `--sizeof--`(), `--subclasshook--`()

40.3.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
<code>args</code> , <code>message</code>	
<i>Inherited from object</i>	
<code>--class--</code>	

40.4 Class Closure

40.4.1 Methods

`--init--`(*self*, *fcn*, **argv*, ***kwarg*)

Create a closure by memorizing a function and its arguments. If all arguments are supplied, you can get the result by calling the closure without arguments. Missing arguments are replaced by instances of class `NotYet`; such arguments will need to be supplied when the closure is called.

As an additional feature, you can supply a list of users of the function's value. When the value is (eventually) produced, the user function(s) will be called with the closure's return value. Users must be functions of one argument.

`--repr--`(*self*)

`--call--`(*self*, **argv*, ***argd*)

This evaluates the closure to either produce a value or raise an exception.

`partial`(*self*, **argv*, ***argd*)

This evaluates the closure to produce another (more complete) closure. Users are not preserved.

`defer`(*self*, *user*)

This will call the specified function when the closure is completed. It's a bit of a kluge.

41 Module *gmisclib.g_datetime*

41.1 Functions

<code>ISO2datetime(<i>s</i>)</code>

<code>timedelta2float(<i>tdel</i>)</code>

41.2 Variables

Name	Description
<code>--package--</code>	Value: 'gmisclib'

42 Module *gmisclib.g_ds9*

Handles interactions with ds9 (<http://hea-www.harvard.edu/RD/ds9/>). Ds9 is an image display and analysis tool commonly used in astrophysics, but it's good for any kind of greyscale image.

Note: there is an extra, special property *Coordsys*. This is not fed to ds9. It is merely something that you can use to remember what coordinate system the regions are in.

42.1 Functions

copy(*r*, ****kwargs**)

Copy a region and update some of the keyword arguments (e.g. *color*). The original object is unchanged, and you can copy any subclass of *Region*.

dequote(*s*)

make_chunks(*s*)

property_parser(*ap*)

test_property_parser()

region_parser(*s*, *defprop*)

test_region_parser()

text_kluge(*s*)

test_text_kluge()

r_list_factory(*slist*, *Coordsys*)

read_regions_iter(*fd*, *Coordsys*=None)

Read regions from a file. This is an iterator. *Fd* is a file descriptor.

read_regions(*fd*, *Coordsys*=None)

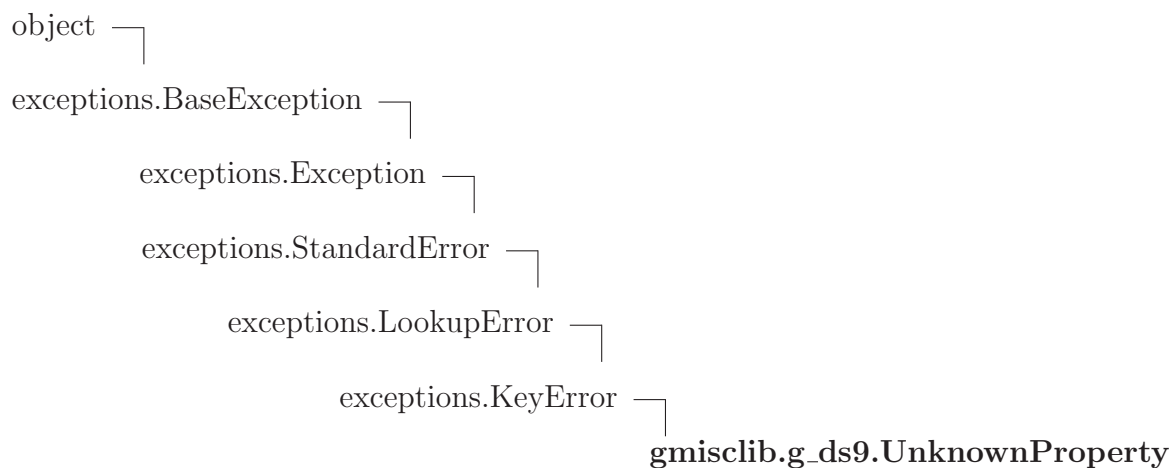
Read regions from a file. *Fd* is a file descriptor.

`test_io()``test_tags()``test_plot()`

42.2 Variables

Name	Description
<code>COLORS</code>	Value: ('white', 'black', 'red', 'green', 'blue', 'cyan', 'magen...
<code>PLOTSYMS</code>	Value: ('circle', 'diamond', 'plus', 'cross')
<code>PROPin</code>	Value: {'color': <type 'str'>, 'delete': <type 'int'>, 'edit': <...
<code>PROPout</code>	Value: {'color': _kvstr, 'text': _kvquote, 'font': _kvquote, 'se...
<code>PLOTSTYLES</code>	Value: ('discrete', 'line', 'step', 'quadratic', 'errorbar')
<code>--package--</code>	Value: 'gmisclib'

42.3 Class `UnknownProperty`



42.3.1 Methods

`__init__`(*self*, **s*)

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` extit(inherited documentation)

Inherited from `exceptions.KeyError`

`__new__`(), `__str__`()

Inherited from `exceptions.BaseException`

`__delattr__`(), `__getattr__`(), `__getitem__`(), `__getslice__`(), `__reduce__`(), `__repr__`(),
`__setattr__`(), `__setstate__`(), `__unicode__`()

Inherited from `object`

`__format__`(), `__hash__`(), `__reduce_ex__`(), `__sizeof__`(), `__subclasshook__`()

42.3.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
<code>args</code> , <code>message</code>	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

42.4 Class `execwait`

```

object └─
          gmisclib.g_ds9.execwait

```

42.4.1 Methods

`__init__`(*self*, *s*)

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` extit(inherited documentation)

`wait`(*self*)

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

42.4.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

42.5 Class `ds9_io`

```

object └─
          gmisclib.g_ds9.ds9_io

```

Known Subclasses: `gmisclib.g_ds9.ds9`

42.5.1 Methods

```
__init__(self, ds9start=None, xpaname='ds9', xaset=['xaset'],
xaset_p=['xaset', '-p'], xpaget=['xpaget'])
```

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` extit(inherited documentation)

```
set(self, *args)
```

```
set_with_input(self, input, *args)
```

Run the `xaset` command and feed it data on its standard input. Each element in the input list becomes one line of data to `xaset`.

Parameters

input: (*type=list or iterator*)

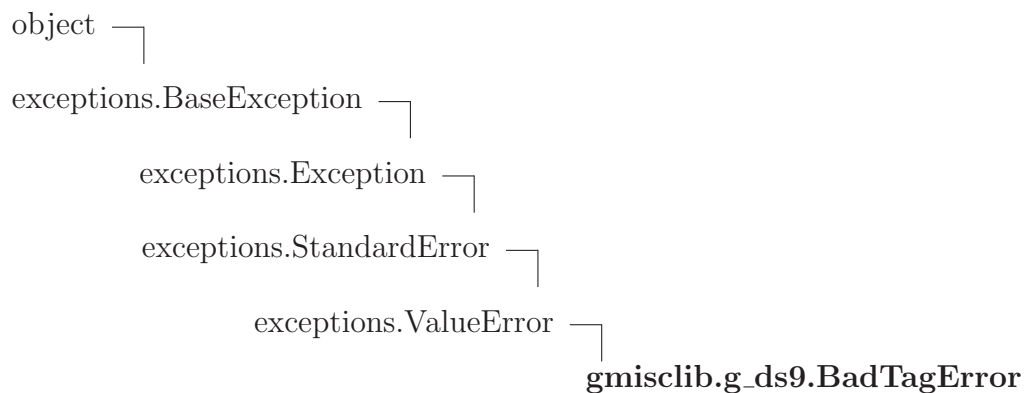
args: The command line used to execute `xaset`.

(*type=["xaset", ...]*)

```
getlast(self, *args)
```

`getall(self, *args)``getiter(self, *args)``close(self)``__del__(self)`***Inherited from `object`***`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`**42.5.2 Properties**

Name	Description
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

42.6 Class `BadTagError`**42.6.1 Methods**`__init__(self, *args)``x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signatureOverrides: `object.__init__` `extit`(inherited documentation)***Inherited from `exceptions.ValueError`***

`__new__()`

Inherited from *exceptions.BaseException*

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

Inherited from *object*

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

42.6.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
args, message	
<i>Inherited from object</i>	
__class__	

42.7 Class Region

object └─
 gmisclib.g_ds9.Region

Known Subclasses: gmisclib.g_ds9.circle, gmisclib.g_ds9.line, gmisclib.g_ds9.point, gmisclib.g_ds9.text

42.7.1 Methods

__init__(*self*, *kwargs*, *coordsys*)
x.**__init__**(...) initializes *x*; see *x*.**__class__**.**__doc__** for signature
 Overrides: *object*.**__init__** extit(inherited documentation)

__str__(*self*)
str(*x*)
 Overrides: *object*.**__str__** extit(inherited documentation)

addprop(*self*, *k*, *v*)

has_prop(*self*, *k*)

```
getprop(self, *args)
```

```
props(self)
```

```
chgprops(self, **kwargs)
```

```
compatible_coord(self, coordsys)
```

Inherited from *object*

```
__delattr__(), __format__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
__repr__(), __setattr__(), __sizeof__(), __subclasshook__()
```

42.7.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

42.8 Class line



42.8.1 Methods

```
__init__(self, x0, y0, x1, y1, Coordsys=None, **kwargs)
x.__init__(...) initializes x; see x.__class__.__doc__ for signature
Overrides: object.__init__ extit(inherited documentation)
```

```
__str__(self)
str(x)
Overrides: object.__str__ extit(inherited documentation)
```

```
end(self, whichend)
```


`center(self)``addprop(self, k, v)``chgprops(self, **kwargs)``compatible_coord(self, coordsys)``getprop(self, *args)``has_prop(self, k)``props(self)`***Inherited from object***

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__subclasshook__()`

42.8.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

42.9 Class *circle***42.9.1 Methods**`__init__(self, x, y, r, Coordsys=None, **kwargs)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` extit(inherited documentation)

`__str__(self)``str(x)`Overrides: `object.__str__` `exitit`(inherited documentation)`center(self)``addprop(self, k, v)``chgprops(self, **kwargs)``compatible_coord(self, coordsys)``getprop(self, *args)``has_prop(self, k)``props(self)`**Inherited from object**`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__subclasshook__()`**42.9.2 Properties**

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

42.10 Class point

42.10.1 Methods

```
__init__(self, x, y, Coordsys=None, **kwargs)
```

x.**__init__**(...) initializes *x*; see *x*.**__class__**.**__doc__** for signature

Overrides: object.**__init__** extit(inherited documentation)

```
__str__(self)
```

str(*x*)

Overrides: object.**__str__** extit(inherited documentation)

```
addprop(self, k, v)
```

```
chgprops(self, **kwargs)
```

```
compatible_coord(self, coordsys)
```

```
getprop(self, *args)
```

```
has_prop(self, k)
```

```
props(self)
```

Inherited from object

```
__delattr__(), __format__(), __getattribute__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),  
__repr__(), __setattr__(), __sizeof__(), __subclasshook__()
```

42.10.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

42.11 Class text



42.11.1 Methods

```
__init__(self, x, y, text, Coordsys=None, **kwargs)
```

x.**__init__**(...) initializes *x*; see *x*.**__class__**.**__doc__** for signature

Overrides: object.**__init__** extit(inherited documentation)

```
__str__(self)
```

str(*x*)

Overrides: object.**__str__** extit(inherited documentation)

```
addprop(self, k, v)
```

```
chgprops(self, **kwargs)
```

```
compatible_coord(self, coordsys)
```

```
getprop(self, *args)
```

```
has_prop(self, k)
```

```
props(self)
```

Inherited from object

__delattr__(), **__format__**(), **__getattr__**(), **__hash__**(), **__new__**(), **__reduce__**(), **__reduce_ex__**(), **__repr__**(), **__setattr__**(), **__sizeof__**(), **__subclasshook__**()

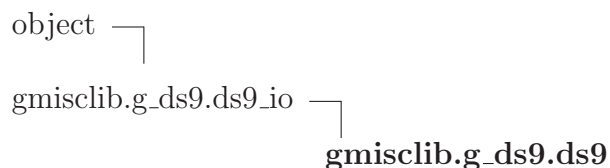
42.11.2 Properties

Name	Description
<i>Inherited from object</i>	

continued on next page

Name	Description
<code>__class__</code>	

42.12 Class *ds9*



42.12.1 Methods

`__init__(self, ds9start=None)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` `exitit`(inherited documentation)

`load(self, fn)`

`get_current_fname(self)`

`get_frames(self)`

`frame_cmd(self, *cmds)`

A catch-all command for anything that's not otherwise implemented.

`del_frames(self, framelist)`

Note: deleting all the frames crashes `ds9`.

`select_frame(self, i)`

Select a frame on the attached `ds9` image display. The frame will be created if it didn't already exist.

Parameters

`i`: Which frame? Apparently, any integer is OK, even negative.

(*type*=**`int`**)

`get_regions_iter(self)`

```
get_regions(self)
```

```
delete_all_regions(self)
```

```
delete_region_group(self, *group)
```

```
add_region(self, *region)
```

Add a list of regions to the current frame.

Parameters

region: one or more regions.

(*type*=[**region**], where **region** is normally an instance of **Region**, but can be anything that produces a string suitable for ds9. See

<http://hea-www.harvard.edu/RD/ds9/ref/xpa.html#regions>.)

```
region_cmd(self, *cmds)
```

A catch-all command for anything that's not otherwise implemented. This does `xpaset -p`, so you cannot use it for region definitions that need multi-line input via `set_with_input`.

```
zoom(self, z)
```

```
scale_cmd(self, *cmds)
```

A catch-all command for anything that's not otherwise implemented.

```
cmap_cmd(self, *cmds)
```

A catch-all command for anything that's not otherwise implemented.

```
set_mode(self, m)
```

```
pan_to(self, x, y, coordsys)
```

```
plot(self, name, xylist, title='', xlabel='', ylabel='', format='(x,y)',  
line='solid', color='black', symbol=None)
```

```
close_plot(self, name)
```

```
list_plots(self)
```

<code>comment(self, s)</code>

<code>flush(self)</code>

<code>__del__(self)</code>

<code>close(self)</code>

<code>getall(self, *args)</code>

<code>getiter(self, *args)</code>

<code>getlast(self, *args)</code>

<code>set(self, *args)</code>

<code>set_with_input(self, input, *args)</code>

Run the *xpaset* command and feed it data on its standard input. Each element in the input list becomes one line of data to *xpaset*.

Parameters

input: (*type=list or iterator*)

args: The command line used to execute *xpaset*.

(*type=["xpaset", ...]*)

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

42.12.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

42.12.3 Class Variables

Name	Description
<code>plotname_ok</code>	Value: <code>re.compile(r'[a-z][a-zA-Z]*')</code>

42.13 Class *ds9_file*



This is just a convenient way to write regions to a file, instead of writing them to a ds9 session. Note that plots and other interactive things are simply no-ops.

42.13.1 Methods

```
__init__(self, fd)
```

x.**__init__**(...) initializes *x*; see *x*.**__class__**.**__doc__** for signature

Overrides: *object*.**__init__** *exitit*(inherited documentation)

```
select_frame(self, i)
```

```
add_region(self, region)
```

```
comment(self, s)
```

```
flush(self)
```

```
plot(self, name, *args, **kwargs)
```

```
list_plots(self)
```

Inherited from object

__delattr__(), **__format__**(), **__getattr__**(), **__hash__**(), **__new__**(), **__reduce__**(), **__reduce_ex__**(), **__repr__**(), **__setattr__**(), **__sizeof__**(), **__str__**(), **__subclasshook__**()

42.13.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

43 Module `gmisclib.g_encode`

This module allows strings to be encoded into a reduced subset. It is designed to work for `avio.py`, and to do a minimal mapping, so that the resulting text is human-readable. It is similar to Quoted-printable encoding, but is not specialized to e-mail limitations and is rather more flexible.

Version: \$Revision: 1.10 \$

43.1 Functions

<code>test()</code>

43.2 Variables

Name	Description
<code>--package--</code>	Value: <code>'gmisclib'</code>

43.3 Class `BadFormatError`



Known Subclasses: `gmisclib.fiatio.ConflictingColumnSpecification`

43.3.1 Methods

<code>__init__(self, *x)</code> <code>x.__init__(...)</code> initializes <code>x</code> ; see <code>x.__class__.__doc__</code> for signature Overrides: <code>object.__init__</code> <code>extit</code> (inherited documentation)

Inherited from `exceptions.Exception`

`__new__()`

Inherited from exceptions.BaseException

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

Inherited from object

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

43.3.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
<code>args</code> , <code>message</code>	
<i>Inherited from object</i>	
<code>__class__</code>	

43.4 Class encoder**43.4.1 Methods**

`__init__(self, allowed=None, notallowed=None, regex=None, eschar='%')`

Note that there are some twiddly points in defining encoders – the `notallowed` and `allowed` arguments need to be thought through carefully, as they are passed into the `re` module as part of a regular expression. Certain characters may give surprising results.

`back(self, x)`

Converts back from a string containing `%xx` escape sequences to an unencoded string.

`fwd(self, x)`

Escapes a string so it is suitable for `a=v;` form. Nonprinting characters, along with `[:#]` are converted to `%xx` escapes (hexadecimal). Non-strings will be converted to strings with `repr()`, and can be fed back into the python interpreter.

44 Module *gmisclib.g_entropy*

This returns the entropy of a probability distribution that produced a given sample.

44.1 Functions

entropy_probs(<i>p</i>)

Entropy of a probability distribution.
--

entropy_vec(<i>p</i>, <i>N</i>=None, <i>F</i>=1.0, <i>Clip</i>=0.01)

Entropy of a frequency distribution <i>p</i> . Here, we assume that <i>p</i> is counts derived from multinomial distributed data; they are not normalized to one.

44.2 Variables

Name	Description
<code>--package--</code>	Value: 'gmisclib'

45 Module *gmisclib.g_exec*

Run a Linux command and capture the result. This module is thread-safe.

Other than the *_raw functions, these functions ignore comments lines (e.g. ones starting with '#'), and if they see a line beginning with an error prefix, they raise an **ExecError** exception. The default error prefix is **ERR**, but you can set it for each call. Likewise, the default comment prefix is **COMMENT**, but you can also set it with the **comment** argument of any call.

The basic idea is that

- you start a subprocess.
- you feed in a bunch of strings (one at a time) to its standard input. (These come from the **input** argument. You need to know, in advance, how much data to feed it.) Note that newlines are not added to the **input** entries, so if you want the strings to be lines, they ought to end in a newline.
- then loop over this:
 - if **perline** is not None, it feeds the subprocess's standard input a string, taken from **perline**. (Note that newlines are not added; you need to include them if you want them, and you probably want them.)
 - it produces an output line
 - repeat
 - keep reading until the subprocess closes its standard output.

When **perline** is empty, the subprocess's standard input is closed. These functions continue reading the subprocess's output as long as it keeps producing.

So, you can do this:

```
get(None, ['pwd']) # "/home/gpk/whatever"
getall(None, ['ls']) # Yields a listing of your directory
get(None, ['sum'], input=['wurgle']) # Same as typing "echo wurgle | sum" to bash
getall(None, ['cat'], input=['wurgle
'], perline=['biffle ']) == ['wurgle ', 'biffle ']
```

45.1 Functions

getiter_raw(*s*, *argv*, *input*=None, *perline*=None, *debug*=False)

Read a list of lines from a subprocess.

Parameters

- s:** the name of the program to execute. (Or `None`, in which case, it is taken from `argv[0]`).
(*type*=`str` or `None`)
- argv:** an array of argument to execute.
(*type*=`list(str)`)
- input:** strings to feed to the subprocess on startup, before the first output is read. These are sent to the subprocess's standard input.
(*type*=array, sequence or iterator, containing strings)
- perline:** is a sequence/iterator of strings to feed in, one at a time, as the subprocess is producing data. (These are fed to the subprocess's standard input, one after each output line that it produces.)
(*type*=array, sequence or iterator, containing strings)
- debug:** Should it print some debugging information?
(*type*=`bool`)

Return Value

a sequence of the lines of output that the program produced.
(Newlines are not removed.)
(*type*=a generator of strings.)

Raises

`ExecError` ZZZ

Note: If `input` or `perline` is badly chosen, one can produce a locked loop of pipes. (Locked loops happen when you have too much stuff waiting to be processed.)

```
getiter(s, argv, input=None, perline=None, debug=False, err='ERR:',  
comment='#')
```

Read a list of lines from a subprocess, after dropping junk like comments. (Comment lines begin with the `comment` argument). Raises an exception if the subprocess returns an error line (the beginning of an error line is specified in the `err` argument).

Parameters

- s:** the name of the program to execute. (Or `None`, in which case, it is taken from `argv[0]`).
(*type=*`str` *or* `None`*)*
- argv:** an array of argument to execute.
(*type=*`list(str)`*)*
- input:** strings to feed to the subprocess on startup, before the first output is read. These are sent to the subprocess's standard input.
(*type=*`array`, *sequence or iterator*, *containing strings*)
- perline:** is a sequence/iterator of strings to feed in, one at a time, as the subprocess is producing data. (These are fed to the subprocess's standard input, one after each output line that it produces.)
(*type=*`array`, *sequence or iterator*, *containing strings*)
- debug:** Should it print some debugging information?
(*type=*`bool`)

Return Value

- a sequence of the lines of output that the program produced. (Newlines are not removed.)
(*type=a generator of strings.*)

Raises

- `ExecError ZZZ`

Note: If `input` or `perline` is badly chosen, one can produce a locked loop of pipes. (Locked loops happen when you have too much stuff waiting to be processed.)

```
get(s, argv=None, input=None, perline=None, debug=False, err='ERR:',  
comment='#')
```

Read a single line from a subprocess, after dropping junk like comments.
Raises an exception if the subprocess produces an error line (see `getiter`,
`ERR`) or no output. See `getiter` for arguments.

Return Value

the subprocess's first output line (after dropping comment lines).

(*type*=*str*)

Raises

`ExecError` if the subprocess produces no output, or an error line.

```
get_raw(s, argv=None, input=None, perline=False, debug=False)
```

Read a single line from a subprocess. (One normally uses this for processes
that are always supposed to produce a single line of output.) See `getiter_raw`
for arguments.

Return Value

the subprocess's first output line.

(*type*=*str*)

Raises

`ExecError` if the subprocess produces no output.

```
getlast(s, argv=None, input=None, perline=False, debug=False,  
err='ERR:', comment='#')
```

Read the last line from a subprocess, after dropping junk like comments.
Raises an exception if the subprocess produces an error line (see `getiter`,
`ERR`) or no output. See `getiter` for arguments.

Return Value

the subprocess's first output line (after dropping comment lines).

(*type*=*str*)

Raises

`ExecError` if the subprocess produces no output, or an error line.

```
getall(s, argv, input=None, perline=None, debug=False, err='ERR:',  
comment='#')
```

Read the text lines produced by a subprocess, after dropping junk like comments. Raises an exception if the process produces an error line (see `getiter`, `ERR`). See `getiter` for arguments.

Return Value

the subprocess's first output line (after dropping comment lines).
(*type*=`list(str)`)

Raises

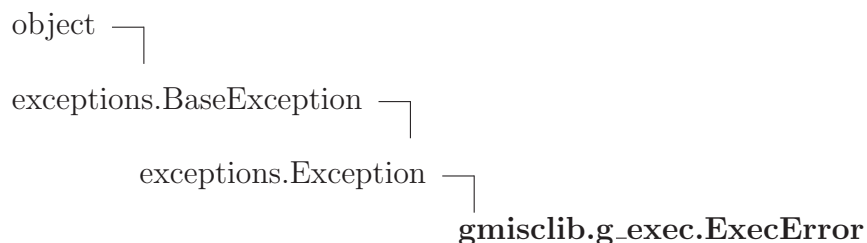
`ExecError` if the process produces an error line.

```
test()
```

45.2 Variables

Name	Description
<code>ERR</code>	Value: 'ERR:'
<code>COMMENT</code>	Value: '#'
<code>__package__</code>	Value: 'gmisclib'

45.3 Class ExecError



Something went wrong. The subprocess did not produce output when some was expected, or it generated an error line (see `ERR`, `err` argument). Perhaps, some of the input that you supplied could not be used. Also, pipe creation might have failed.

45.3.1 Methods

`__init__(self, *s)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` `exitit`(inherited documentation)

Inherited from `exceptions.Exception`

`__new__()`

Inherited from `exceptions.BaseException`

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

Inherited from `object`

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

45.3.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
<code>args</code> , <code>message</code>	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

46 Module *gmisclib.g_implements*

This module tells you if an object's signature matches a class. It lets you know if one class can substitute for another, even if they are unrelated in an OO hierarchy.

This allows you to begin to check that a foreign object will have the necessary properties to be used in your code. It is more of a functional check than `isimplements()`.

For instance, if your code needs a `write()` method, you can check for it like this:

```
class some_template(object):
    def write(self, somearg):
        pass

#
g_implements.check(x, some_template)
#... at this point, we can be assured that a
# call to x.write(3) is possible.
# (Note that we don't know what the call will *do*,
# merely that the proper method of x exists and that
# it can take an argument.
```

Semi-kluge: if you have an instance, `i` that needs to convince `g_implements` that it can actually behave as class `C`, then set `i.__g_implements__ = ['C']`. Generally, this can be a list or set of class names. This trick is useful for C-implementations of python classes (where introspection cannot get all the necessary information) or for fancy python classes that use `__getattr__` and similar.

46.1 Functions

Vartype (<i>mem</i> , <i>com</i>)
--

Strict (<i>mem</i> , <i>com</i>)

why (<i>instance</i> , <i>classobj</i>)
--

Explains why instance doesn't implement the classobj.

Parameters

instance: an instance of a class

classobj: a class object

Return Value

None (if it does) or a string explanation of why it doesn't @rtype

None or str

impl(*instance*, *classobj*)

Tells you if an instance of an object implements a class. By implements, I mean that the instance supplies every member that the class supplies, and every member has the same type. The instance may have **more** members, of course. Functions require that the argument names must match, too at least as far as the required arguments in the classobj's function.

The match may be made looser by adding a `g_implements` attribute to various class members. Possibilities for the value are `Optional`, `Strict`, `Vartype`, `Varargs`, or you can give a two-argument function, and that function will be called to decide whether the match is acceptable or not.

make_optional(*x*)

This is a decorator for a function. `make_optional` implies `make_varargs`.

make_varargs(*x*)

This is a decorator for a function.

make_vartype(*x*)

This is a decorator for a function.

make_strict(*x*)

This is a decorator for a function.

check(*instance*, *classobj*)

Require that **instance** has the attributes defined by **classobj**. This function either quietly succeeds or it raises a `GTypeError` exception.

Parameters

instance: the thing to check

(type=any instance of a class (i.e. anything with a `--class--` attribute).)

classobj: a class that provides a pattern of attributes.

(type=any class object (i.e. not an instance, typically).)

Raises

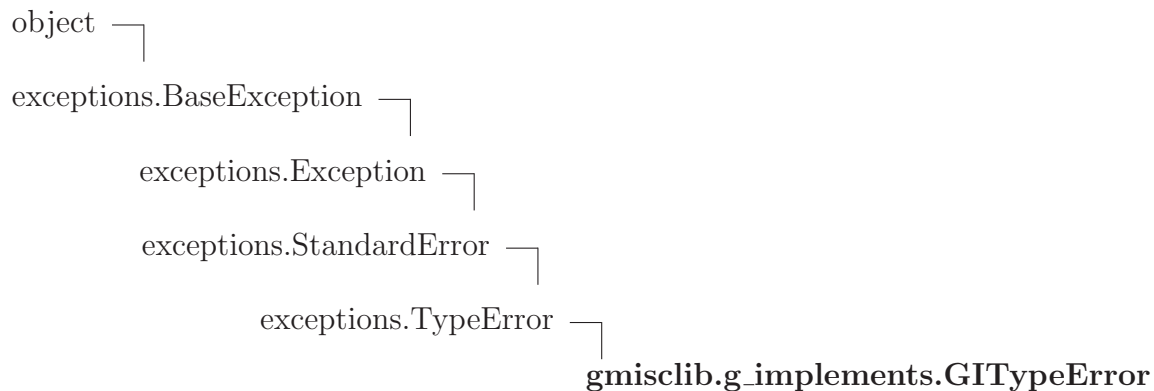
`GTypeError` **instance** doesn't provide all the features of **classobj**.

test()

46.2 Variables

Name	Description
<code>MAXCACHE</code>	Value: 100
<code>Optional</code>	Value: 'optional'
<code>Varargs</code>	Value: 'varargs'
<code>Ignore</code>	Value: set(['__class__', '__delattr__', '__dict__', '__doc__', '...'])
<code>__package__</code>	Value: 'gmisclib'

46.3 Class `GTypeError`



Exception raised by `check` to indicate failure.

46.3.1 Methods

```

__init__(self, s)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature
Overrides: object.__init__ extit(inherited documentation)

```

Inherited from `exceptions.TypeError`

```
__new__()
```

Inherited from `exceptions.BaseException`

```

__delattr__(), __getattr__(), __getitem__(), __getslice__(), __reduce__(), __repr__(),
__setattr__(), __setstate__(), __str__(), __unicode__()

```

Inherited from `object`

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

46.3.2 Properties

Name	Description
	<i>Inherited from <code>exceptions.BaseException</code></i>
	args, message
	<i>Inherited from object</i>
<code>__class__</code>	

47 Module `gmisclib.g_keyed_accum`

47.1 Functions

<code>test()</code>

47.2 Variables

Name	Description
<code>--package--</code>	Value: <code>'gmisclib'</code>

47.3 Class `keyed_avg`

`gmisclib.g_keyed_accum.keyed_thing` — `gmisclib.g_keyed_accum.keyed_avg`

47.3.1 Methods

<code>--init__(self)</code>

Overrides: `gmisclib.g_keyed_accum.keyed_thing.__init__`

<code>add(self, key, value)</code>

Inherited from `gmisclib.g_keyed_accum.keyed_thing`

`--contains--()`, `--delitem--()`, `--iter--()`, `--len--()`, `clear()`, `get()`, `has_key()`, `items()`, `iterkeys()`, `keys()`, `values()`

48 Module *gmisclib.g_lin_fit*

Generic linear best fits.

48.1 Functions

fit(*idata*, *fitting_fcns*, *ff_info*=None, *wt*=None)

Fits data to a linear combination of fitting_fcns.

Parameters

fitting_fcns: the basis functions for the linear regression. The fitting functions return vectors (which must match the data). Fitting functions are called as *f*(*info*, *ff_info*, *len*(vector)). Vectors may have different lengths, as long as the corresponding data and fit vectors agree.

(type=an array of functions.)

idata: *(type=an array of (vector, info).)*

Return Value

The returned value is a tuple of: the coefficients of the optimal solution (as an array), the residual (a single float number), the rank of the fit (int), an array of the singular values, and the best fit to the data. The returned value follows Num.LA.linear_least_squares(), except that residual is always calculated and is a scalar.

(type=tuple)

test()

test1()

test2()

test3()

test4()

test5()

test6()

48.2 Variables

Name	Description
<code>--package--</code>	Value: 'gmisc.lib'

49 Module *gmisclib.g_localfit*

Fit a linear transform to a bunch of *tt* input/output vectors.

49.1 Functions

err_before_fit(*data*, *minsv*=None, *minsvr*=None, *constant*=True)

How much variation did the data have before the fit? This is used to compare with the error after the fit, to allow a F-test or ANOVA. Strictly speaking, we compute the mean-squared error about a constant.

Parameters

data: See *pack*.

(*type*=See *pack*.)

Return Value

Returns the per-coordinate sum-squared-error of the output coordinates.

(*type*=*numpy.ndarray*)

pack(*data*, *constant*=True)

Prepare the data array and (optionally) weight the data.

Parameters

data: [(*input_coordinates*, *output_coordinates*), ...] or [(*input_coordinates*, *output_coordinates*, *weight*), ...]. A list of (in,out) tuples corresponding to the independent and dependent parameters of the linear transform. Both **in** and **out** are one-dimensional *numpy* vectors. They don't need to have the same length, though (obviously) all instances of "in" need to have the same length and all instances of "out" also need to match one another. If **weight** is given, it must be a scalar.

(*type*=[(*numpy.ndarray*, *numpy.ndarray*), ...] or [(*numpy.ndarray*, *numpy.ndarray*, *float*), ...])

localfit(*data*, *minsv*=None, *minsvr*=None, *constant*=True)

Does a linear fit to data via a singular value decomposition algorithm. It returns the matrix A and vector B such that $A * \text{input_coordinates} + B$ is the best fit to `output_coordinates`.

Parameters

- minsv:** sets the minimum useable singular value;
(*type*=float or None)
- minsvr:** sets the minimum useable s.v. in terms of the largest s.v.
(*type*=float or None)
- constant:** Do you want a constant built into the linear equations?
(*type*=Boolean)
- data:** list of tuples. See `pack`.
(*type*=See ***pack***.)

Return Value

(A, B, errs, sv, ranks) where

- A is a 2-D `numpy.ndarray` matrix.
- B is a 1-D `numpy.ndarray` vector (if constant, else None).
- errs is a `numpy.ndarray` vector, one value for each output coordinate. It gives the sum of squared residuals.
- sv are the singular values, sorted into decreasing order.

reg_localfit(*data*, *regstr*=0.0, *regtgt*=None, *rscaler*=None, *constant*=True)

Does a linear fit to data via a singular value decomposition algorithm. It returns the matrix A and vector B such that $A \cdot \text{input_coordinates} + B$ is the best fit to *output_coordinates*.

Parameters

constant: Do you want a constant built into the linear equations?

(*type*=*Boolean*)

data: list of tuples. See **pack**.

(*type*=*See pack.*)

Return Value

(A, B, errs, sv, ranks) where

- A is a 2-D `numpy.ndarray` matrix.
- B is a 1-D `numpy.ndarray` vector (if constant, else `None`).
- errs is a `numpy.ndarray` vector, one value for each output coordinate. It gives the sum of squared residuals.
- sv are the singular values, sorted into decreasing order.

robust_localfit(*data*, *minsv*=None, *minsvr*=None, *constant*=True)

Data is [(input_coordinates, output_coordinates), ...] *Minsv* sets the minimum useable singular value; *minsvr* sets the minimum useable s.v. in terms of the largest s.v.. It returns the matrix A and vector B such that the best fit is $A \cdot \text{input_coordinates} + B$ in a tuple (A, B, errs, ranks). *errs* is a vector, one value for each output coordinate. Rank is the minimum rank of the various fits.

Warning! Not tested.

fit_giving_sigmas(*data*, *minsv*=None, *minsvr*=None, *constant*=True)

Does a linear fit to data via a singular value decomposition algorithm. It returns the matrix A and vector B such that $A \cdot \text{input_coordinates} + B$ is the best fit to `output_coordinates`.

Parameters

- minsv:** sets the minimum useable singular value;
(*type*=float or None)
- minsvr:** sets the minimum useable s.v. in terms of the largest s.v.
(*type*=float or None)
- constant:** Do you want a constant built into the linear equations?
(*type*=Boolean)
- data:** [(input_coordinates, output_coordinates), ...]. A list of (in,out) tuples corresponding to the independent and dependent parameters of the linear transform. Both `in` and `out` are one-dimensional **numpy vectors**. They don't need to have the same length, though (obviously) all instances of "in" need to have the same length and all instances of "out" also need to match one another.
(*type*=[(numpy.ndarray, numpy.ndarray), ...])

Return Value

- (A, B, sigmaA, sigmaB) where
- A is a 2-D `numpy.ndarray` matrix.
 - B is a 1-D `numpy.ndarray` vector (if constant, else None).)

leaktest()

test_localfit11(*r*)

test_localfit11e()

test_localfit21()

test_localfit21u()

test_localfit22(*r*)

test_fgs1()

`test_wt()`

49.2 Variables

Name	Description
--package--	Value: 'gmiscLib'

50 Module *gmisclib.g_lpc*

Linear prediction of signals

50.1 Functions

lp1_fit (<i>y, x, indices, noise=0.0</i>)
--

Predicts $y[i]$ in terms of the set of $x[i+q]$ for q in <i>indices</i> .

add_sloppy (<i>a, b, offset</i>)

Accumulate b onto a . Allow b to be smaller (fill with zeros), or larger (drop).
--

lp1_run (<i>x, coef, indices</i>)
--

51 Module gmisclib.g_pipe

A multithreaded version of `os.popen2()`. Note that the argument list isn't quite the same.

51.1 Functions

popen2(*path*, *args*, *bufsize*=0)

Forks off a process, and returns the processes input and output file descriptors. The latter is really a pfd (defined above).

Path and args are passed directly into `os.execvp()`.

test()

51.2 Variables

Name	Description
<code>__package__</code>	Value: 'gmisclib'

51.3 Class pfd

object —
 gmisclib.g_pipe.pfd

Pseudo-file descriptor. Just like a FD, except that it waits for the process at the end of the pipe to terminate when you close it.

51.3.1 Methods

__init__(*self*, *p*)

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` extit(inherited documentation)

read(*self*)

readline(*self*)

`readlines(self)``next(self)``__iter__(self)``flush(self)``xreadlines(self, sizehint=-1)``close(self)``__del__(self)`***Inherited from object***

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

51.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

51.3.3 Class Variables

Name	Description
<code>__doc__</code>	Value: <code>"""Pseudo-file descriptor. Just like a FD, exc...</code>

52 Module *gmisclib.g_pipe_old*

A multithreaded version of `os.popen2()`. Note that the argument list isn't quite the same.

52.1 Functions

popen2(*path*, *args*)

Forks off a process, and returns the processes input and output file descriptors. The latter is really a *pfid* (defined above).

Path and args are passed directly into `os.execvp()`.

test()

52.2 Variables

Name	Description
<code>__package__</code>	Value: <code>'gmisclib'</code>

52.3 Class *pfid*

object —
 gmisclib.g_pipe_old.pfid

Pseudo-file descriptor. Just like a FD, except that it waits for the process at the end of the pipe to terminate when you close it.

52.3.1 Methods

__init__(*self*, *fd*, *cpid*)

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` `exitit` (inherited documentation)

read(*self*)

readline(*self*)

`readlines(self)``next(self)``__iter__(self)``flush()``xreadlines(self, sizehint=-1)``close(self)``__del__(self)`***Inherited from object***

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

52.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

52.3.3 Class Variables

Name	Description
<code>__doc__</code>	Value: <code>"""Pseudo-file descriptor. Just like a FD, exc...</code>

53 Module gmisclib.g_place_label

53.1 Functions

Gauss (<i>y</i> , <i>ybar</i> , <i>yvar</i>)

gauss (<i>y</i> , <i>ybar</i> , <i>yvar</i>)

q_gauss (<i>y</i> , <i>ybar</i> , <i>yvar</i>)

53.2 Variables

Name	Description
<code>--package--</code>	Value: 'gmisclib'

53.3 Class boxc

object —
 gmisclib.g_place_label.boxc

Known Subclasses: gmisclib.g_place_label.viewport

53.3.1 Methods

--init-- (<i>self</i> , <i>xc</i> , <i>yc</i> , <i>xw</i> , <i>yw</i>)

x.**--init--**(...) initializes *x*; see *x*.**--class--**.**--doc--** for signature

Overrides: object.**--init--** extit(inherited documentation)

xmin (<i>self</i>)

xmax (<i>self</i>)

ymin (<i>self</i>)

ymax (<i>self</i>)

`w(self)``h(self)``yc(self)``xc(self)``shift(self, x, y)`***Inherited from object***

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

53.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

53.4 Class `text_template`

```

object └─ gmisclib.g_place_label.text_template

```

53.4.1 Methods`__init__(self, text, H, W)`

(0,0) is the lower left corner of the block of text.

Overrides: `object.__init__``addline(self, x0, y0, xlength, yht, txtline)`

Specify LLC.

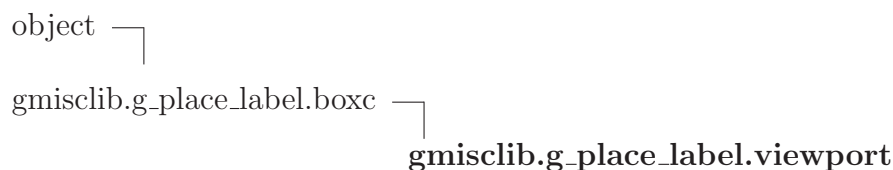
`nlines(self)`***Inherited from object***

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

53.4.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

53.5 Class viewport



53.5.1 Methods

`__init__(self, fracCharHt, fracCharW, xmin, xmax, ymin, ymax)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` extit(inherited documentation)

`dataset(self, x, y)`

`find_y(self, x0, ttm)`

Figures out a good place to put a label. The label is at a specified `tc`, but can be at any vertical position. This finds a good vertical position. Returns the bottom of the block of text.

`textplace(self, x0, text)`

`draw_hor_line(self, xc, yc, w)`

`h(self)`

`shift(self, x, y)`

`w(self)``xc(self)``xmax(self)``xmin(self)``yc(self)``ymax(self)``ymin(self)`***Inherited from object***

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

53.5.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

54 Module `gmisclib.g_pylab`

This class works in concert with `bin/pylab_server.py` to allow you to display simple `pylab/matplotlib` graphics on another machine. It's most useful if you need to do plots from a machine where `pylab` isn't or cannot be installed.

55 Module gmisclib.g_selector

55.1 Functions

```
selector_not(selector)
```

```
test()
```

55.2 Variables

Name	Description
<code>--package--</code>	Value: 'gmisclib'

55.3 Class selector

Known Subclasses: gmisclib.g_selector.false, gmisclib.g_selector.one_selector, gmisclib.g_selector.selector, gmisclib.g_selector.true

55.3.1 Methods

```
accept(self, item)
```

```
--init--(self)
```

```
and_another(self, txt)
```

```
select(self, txt)
```

```
or_another(self, txt)
```

```
And(self, s)
```

```
Or(self, s)
```

```
Not(self)
```

```
leaf(self, txt)
```


filterlist(*self*, *x*)

55.4 Class *true*

gmisclib.g_selector.selector └─ **gmisclib.g_selector.true**

55.4.1 Methods

__init__(*self*)

Overrides: gmisclib.g_selector.selector.__init__

accept(*self*, *item*)

Overrides: gmisclib.g_selector.selector.accept

And(*self*, *s*)

Not(*self*)

Or(*self*, *s*)

and_another(*self*, *txt*)

filterlist(*self*, *x*)

leaf(*self*, *txt*)

or_another(*self*, *txt*)

select(*self*, *txt*)

55.5 Class *false*

gmisclib.g_selector.selector └─ **gmisclib.g_selector.false**

55.5.1 Methods

`__init__`(*self*)Overrides: `gmisclib.g_selector.selector.__init__`**`accept`**(*self*, *item*)Overrides: `gmisclib.g_selector.selector.accept`**`And`**(*self*, *s*)**`Not`**(*self*)**`Or`**(*self*, *s*)**`and_another`**(*self*, *txt*)**`filterlist`**(*self*, *x*)**`leaf`**(*self*, *txt*)**`or_another`**(*self*, *txt*)**`select`**(*self*, *txt*)

55.6 Class `selector_op`

`gmisclib.g_selector.selector``gmisclib.g_selector.selector_op`

55.6.1 Methods

`__init__`(*self*, *l*, *op*, *r*)Overrides: `gmisclib.g_selector.selector.__init__`**`accept`**(*self*, *x*)Overrides: `gmisclib.g_selector.selector.accept`

`And(self, s)``Not(self)``Or(self, s)``and_another(self, txt)``filterlist(self, x)``leaf(self, txt)``or_another(self, txt)``select(self, txt)`

55.7 Class `one_selector`

`gmisclib.g_selector.selector` └─ `gmisclib.g_selector.one_selector`

55.7.1 Methods

`__init__(self, txt)`Overrides: `gmisclib.g_selector.selector.__init__``accept(self, item)`Overrides: `gmisclib.g_selector.selector.accept``And(self, s)``Not(self)``Or(self, s)``and_another(self, txt)`

```
filterlist(self, x)
```

```
leaf(self, txt)
```

```
or_another(self, txt)
```

```
select(self, txt)
```

56 Module *gmisclib.gpk_getopt*

56.1 Functions

parse(*arglist*, *shortoptions*, *longoptions*)

Splits apart an command line argument list, and puts the arguments into a dictionary. The arguments are a list to be parsed, a string of short one character options, and a list of long gnu-style options. The function returns a dictionary mapping between option names and values (if any). Names in the dictionary have had leading dashes removed.

56.2 Variables

Name	Description
<code>__package__</code>	Value: <code>'gmisclib'</code>

57 Module *gmisclib.gpk_hdr*

57.1 Functions

<code>uq(<i>s</i>, <i>qc</i>)</code>

57.2 Variables

Name	Description
<code>--package--</code>	Value: <code>'gmisclib'</code>

57.3 Class *hdr*

57.3.1 Methods

<code>--init--(<i>self</i>, <i>fn</i>)</code>

<code>has_key(<i>self</i>, <i>key</i>)</code>

<code>items(<i>self</i>)</code>

<code>get(<i>self</i>, <i>key</i>, <i>default</i>=None, <i>qc</i>=None)</code>
--

<code>set(<i>self</i>, <i>key</i>, <i>value</i>, <i>qc</i>=None)</code>

<code>--del--(<i>self</i>)</code>

<code>close(<i>self</i>)</code>

<code>write(<i>self</i>)</code>

<code>read(<i>self</i>)</code>

58 Module `gmisclib.gpk_lapack`

58.1 Functions

<code>solve(<i>a</i>, <i>b</i>)</code>
--

solves $a \cdot x = b$.

<code>test1()</code>

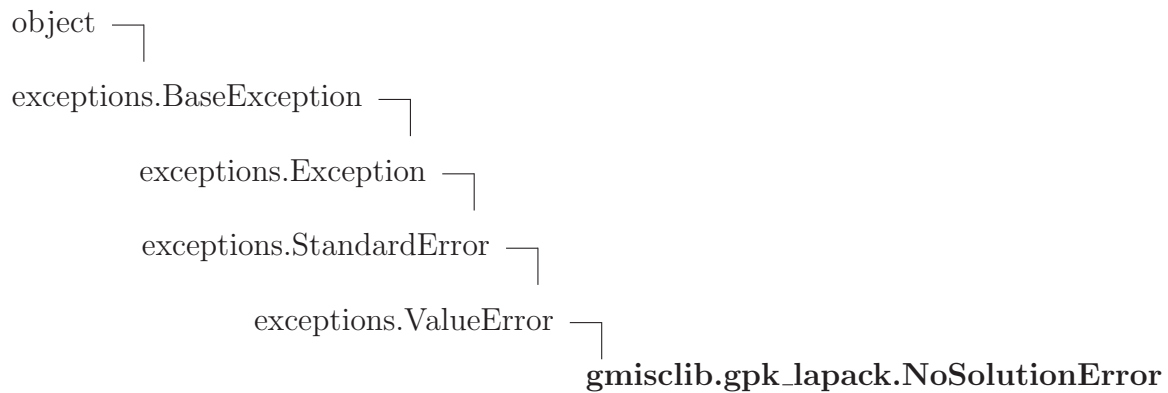
<code>err(<i>a</i>, <i>b</i>)</code>

<code>test2()</code>

<code>testa()</code>

<code>testbdi()</code>

58.2 Class `NoSolutionError`



58.2.1 Methods

<code>__init__(<i>self</i>, <i>s</i>)</code>
--

<code>x.__init__(...)</code> initializes <code>x</code> ; see <code>x.__class__.__doc__</code> for signature
--

Overrides: <code>object.__init__</code> <code>exitit</code> (inherited documentation)

Inherited from `exceptions.ValueError`

`__new__()`

Inherited from exceptions.BaseException

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

Inherited from object

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

58.2.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
args, message	
<i>Inherited from object</i>	
__class__	

59 Module *gmisclib.gpk_lsq*

59.1 Functions

<code>test_svd()</code>

<code>all_between(<i>a</i>, <i>vec</i>, <i>b</i>)</code>
--

<code>test_vec()</code>

<code>test_vec2()</code>

<code>test_m1()</code>

<code>test_m2()</code>

<code>test_m2r()</code>

<code>test_m2rR()</code>

<code>test_hat()</code>

Make sure that the hat function meets its definition.

59.2 Variables

Name	Description
<code>--package--</code>	Value: <code>'gmisclib'</code>

59.3 Class *lls_base*

object  `gmisclib.gpk_lsq.lls_base`

Known Subclasses: `gmisclib.gpk_lsq.linear_least_squares`, `gmisclib.gpk_lsq.reg_linear_least_squares`

59.3.1 Methods

`__init__(self, a, copy=True)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` extit(inherited documentation)

`set_y(self, y, copy=True)`

`y(self, copy=True)`

`hat(self, copy=True)`

`x(self, y=None, copy=True)`

`fit(self, copy=False)`

`residual(self)`

`variance_about_fit(self)`

Returns the estimator of the standard deviation of the data about the fit.

Return Value

`numpy.ndarray` with `shape=(q,)`. Each entry corresponds to one of the `q` sets of equations that are being fit.

`eff_rank(self)`

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

59.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

59.4 Class `linear_least_squares`



Known Subclasses: `gmisclib.gpk_rlsq.w_linear_least_squares`

59.4.1 Methods

`__init__(self, a, y=None, minsv=None, minsvr=None, copy=True)`

This solves the set of linear equations $a \cdot x = y$, and returns $(x, \text{the_fit}, \text{rank}, s)$. Normally, $a.\text{shape} == (m, n)$ and $y.\text{shape} == (m, q)$, and the returned $x.\text{shape} == (n, q)$. where m is the number of constraints provided by the data, n is the number of parameters to use in a fit (equivalently, the number of basis functions), and q is the number of separate sets of equations that you are fitting. Then, x has shape (n, q) and the_fit has shape (m, q) . Interpreting this as a linear regression, there are n parameters in the model, and m measurements. Q is the number of times you apply the model to a different data set; each one yields a different solution.

The procedure uses a singular value decomposition algorithm, and treats all singular values that are smaller than `minsv` as zero (i.e. drops them). If `minsvr` is specified, it treats all singular values that are smaller than `minsvr` times the largest s.v. as zero. The returned rank is the rank of the solution, which is normally the number of nonzero elements in x . Note that the shape of the solution vector or matrix is defined by a and y , and the rank can be smaller than m .

Y may be a 1-D matrix (a vector), in which case the fit is a vector. This is the normal case where you are fitting one equation. If y is a 2-D matrix, each column (second index) in y is a separate fit, and each column in the solution is a separate result.

Overrides: `object.__init__`

`sv(self)`

`rank(self)`

hat(*self*, *copy*=True)

Hat Matrix Diagonal Data points that are far from the centroid of the X-space are potentially influential. A measure of the distance between a data point, x_i , and the centroid of the X-space is the data point's associated diagonal element h_i in the hat matrix. Belsley, Kuh, and Welsch (1980) propose a cutoff of $2p/n$ for the diagonal elements of the hat matrix, where n is the number of observations used to fit the model, and p is the number of parameters in the model. Observations with h_i values above this cutoff should be investigated. For linear models, the hat matrix

$$H = X \text{ inv}(X'X) X'$$

can be used as a projection matrix. The hat matrix diagonal variable contains the diagonal elements of the hat matrix

$$h_i = x_i \text{ inv}(X'X) x_i'$$

Overrides: `gmisclib.gpk_lsq.lls_base.hat`

x_variances(*self*)

Estimated standard deviations of the solution. This is the diagonal of the solution covariance matrix.

eff_rank(*self*)

fit(*self*, *copy*=False)

residual(*self*)

set_y(*self*, *y*, *copy*=True)

variance_about_fit(*self*)

Returns the estimator of the standard deviation of the data about the fit.

Return Value

`numpy.ndarray` with shape=(q). Each entry corresponds to one of the q sets of equations that are being fit.

x(*self*, *y*=None, *copy*=True)

y(*self*, *copy*=True)

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

59.4.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

59.5 Class `reg_linear_least_squares`



59.5.1 Methods

<code>__init__(self, a, y, regstr=0.0, regtgt=0.0, rscale=None, copy=True)</code>
<p>This solves $\min \ a \cdot x - y\ ^2 + \text{regstr} \cdot (x - \text{regtgt}) ^2$, and returns $(x, \text{the_fit}, \text{rank}, s)$. Normally, $a.\text{shape} == (m, n)$ and $y.\text{shape} == (m, q)$, where m is the number of data to be fit, n is the number of parameters to use in a fit (equivalently, the number of basis functions), and q is the number of separate sets of equations that you are fitting. Then, x has shape (n, q) and the_fit has shape (m, q).</p> <p>The regularization target, regtgt is the same shape as x, that is (n, q). (It must be a vector if and only if y is a vector.) Regstr, the strength of the regularization is normally an (n, n) matrix, though $(*, n)$ will work, as will a scalar.</p> <p>Y may be a 1-D matrix (a vector), in which case the fit is a vector. This is the normal case where you are fitting one equation. If y is a 2-D matrix, each column (second index) in y is a separate fit, and each column in the solution is a separate result.</p> <p>Overrides: <code>object.__init__</code></p>

<code>sv_reg(self)</code>
Singular values of the regularized problem.

sv_unreg(*self*)

Singular values of the unregularized problem.

hat(*self*, *copy=True*)

Hat Matrix Diagonal Data points that are far from the centroid of the X-space are potentially influential. A measure of the distance between a data point, x_i , and the centroid of the X-space is the data point's associated diagonal element h_i in the hat matrix. Belsley, Kuh, and Welsch (1980) propose a cutoff of $2p/n$ for the diagonal elements of the hat matrix, where n is the number of observations used to fit the model, and p is the number of parameters in the model. Observations with h_i values above this cutoff should be investigated. For linear models, the hat matrix

$$H = X (X'X)^{-1} X'$$

can be used as a projection matrix. The hat matrix diagonal variable contains the diagonal elements of the hat matrix

$$h_i = x_i (X'X)^{-1} x_i'$$

Overrides: *gmisclib.gpk_lsq.lls_base.hat*

eff_rank(*self*)**fit**(*self*, *copy=False*)**residual**(*self*)**set_y**(*self*, *y*, *copy=True*)**variance_about_fit**(*self*)

Returns the estimator of the standard deviation of the data about the fit.

Return Value

`numpy.ndarray` with shape=(q). Each entry corresponds to one of the q sets of equations that are being fit.

x(*self*, *y=None*, *copy=True*)**y**(*self*, *copy=True*)*Inherited from object*

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

59.5.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

60 Module *gmisclib.gpk_rlsq*

60.1 Functions

<code>test_w1()</code>

<code>test_w1b()</code>

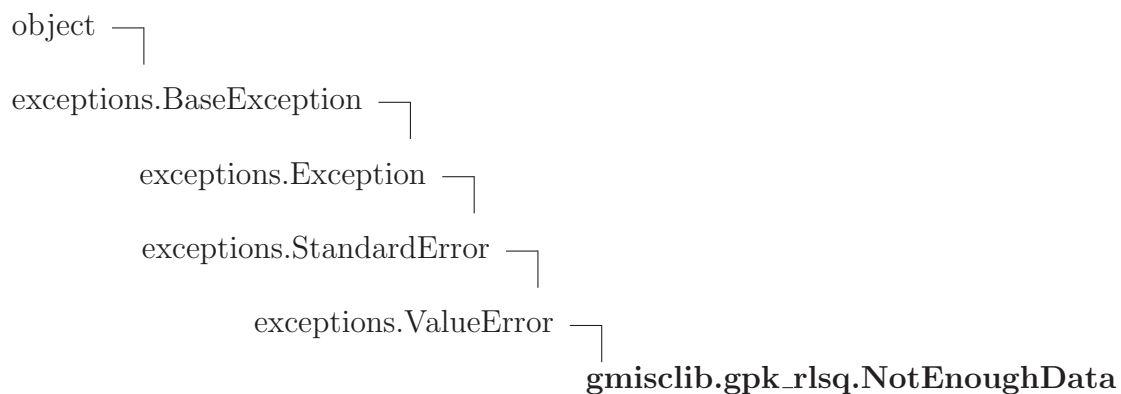
<code>test_r1()</code>

<code>test_r1hat()</code>

60.2 Variables

Name	Description
<code>__package__</code>	Value: <code>'gmisclib'</code>

60.3 Class *NotEnoughData*



60.3.1 Methods

<code>__init__(self, s)</code>

<code>x.__init__(...)</code> initializes x; see <code>x.__class__.__doc__</code> for signature
--

Overrides: <code>object.__init__</code> <code>exitit</code> (inherited documentation)

Inherited from `exceptions.ValueError`

`__new__()`

Inherited from `exceptions.BaseException`

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

Inherited from `object`

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

60.3.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
<code>args</code> , <code>message</code>	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

60.4 Class `w_linear_least_squares`



Known Subclasses: `gmisclib.gpk_rlsq.robust_linear_fit`

60.4.1 Methods

`__init__(self, a, y, w, minsv=None, minsvr=None, copy=True)`

This solves the set of linear equations $a \cdot x = y$, and returns $(x, \text{the_fit}, \text{rank}, s)$. Normally, $a.\text{shape} == (m, n)$ and $y.\text{shape} == (m, q)$, and the returned $x.\text{shape} == (n, q)$. where m is the number of constraints provided by the data, n is the number of parameters to use in a fit (equivalently, the number of basis functions), and q is the number of separate sets of equations that you are fitting. Then, x has shape (n, q) and the_fit has shape (m, q) . Interpreting this as a linear regression, there are n parameters in the model, and m measurements. Q is the number of times you apply the model to a different data set; each on yields a different solution.

The procedure uses a singular value decomposition algorithm, and treats all singular values that are smaller than `minsv` as zero (i.e. drops them). If `minsvr` is specified, it treats all singular values that are smaller than `minsvr` times the largest s.v. as zero. The returned rank is the rank of the solution, which is normally the number of nonzero elements in x . Note that the shape of the solution vector or matrix is defined by a and y , and the rank can be smaller than m .

Y may be a 1-D matrix (a vector), in which case the fit is a vector. This is the normal case where you are fitting one equation. If y is a 2-D matrix, each column (second index) in y is a separate fit, and each column in the solution is a separate result.

Overrides: `object.__init__` `exitit`(inherited documentation)

`fit(self, copy=False)`

Overrides: `gmisclib.gpk_lsq.lls_base.fit`

`y(self, copy=True)`

Overrides: `gmisclib.gpk_lsq.lls_base.y`

`set_y(self, y, copy=True)`

Overrides: `gmisclib.gpk_lsq.lls_base.set_y`

`residual(self)`

Overrides: `gmisclib.gpk_lsq.lls_base.residual`

`eff_rank(self)`

hat(*self*, *copy*=True)

Hat Matrix Diagonal Data points that are far from the centroid of the X-space are potentially influential. A measure of the distance between a data point, x_i , and the centroid of the X-space is the data point's associated diagonal element h_i in the hat matrix. Belsley, Kuh, and Welsch (1980) propose a cutoff of $2p/n$ for the diagonal elements of the hat matrix, where n is the number of observations used to fit the model, and p is the number of parameters in the model. Observations with h_i values above this cutoff should be investigated. For linear models, the hat matrix

$$H = X \text{ inv}(X'X) X'$$

can be used as a projection matrix. The hat matrix diagonal variable contains the diagonal elements of the hat matrix

$$h_i = x_i \text{ inv}(X'X) x_i'$$

Overrides: `gmisclib.gpk_olsq.ols_base.hat`

rank(*self*)

sv(*self*)

variance_about_fit(*self*)

Returns the estimator of the standard deviation of the data about the fit.

Return Value

`numpy.ndarray` with shape=(q). Each entry corresponds to one of the q sets of equations that are being fit.

x(*self*, *y*=None, *copy*=True)

x_variances(*self*)

Estimated standard deviations of the solution. This is the diagonal of the solution covariance matrix.

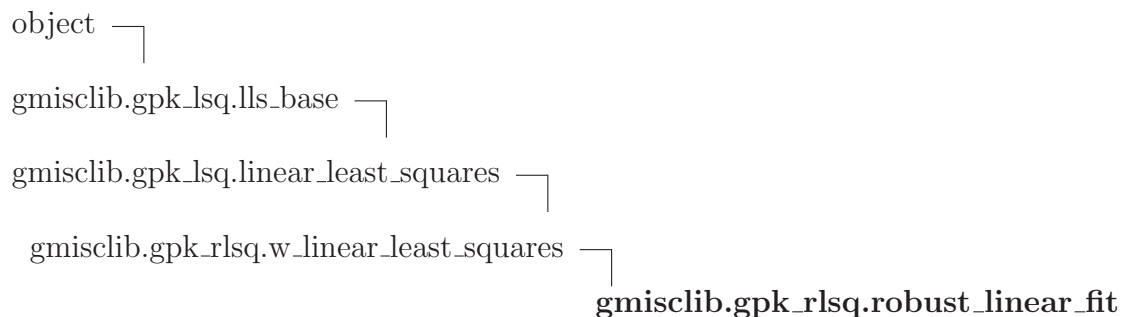
Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

60.4.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

60.5 Class *robust_linear_fit*



60.5.1 Methods

`__init__(self, a, y, sigma_y=None, minsv=None, minsvr=None, copy=True)`

This does bounded influence regression, with a robust M-estimator in the y-direction.

Overrides: `object.__init__`

`eff_rank(self)`

`fit(self, copy=False)`

Overrides: `gmisclib.gpk_lsq.lls_base.fit`

hat(*self*, *copy*=True)

Hat Matrix Diagonal Data points that are far from the centroid of the X-space are potentially influential. A measure of the distance between a data point, x_i , and the centroid of the X-space is the data point's associated diagonal element h_i in the hat matrix. Belsley, Kuh, and Welsch (1980) propose a cutoff of $2p/n$ for the diagonal elements of the hat matrix, where n is the number of observations used to fit the model, and p is the number of parameters in the model. Observations with h_i values above this cutoff should be investigated. For linear models, the hat matrix

$$H = X \text{ inv}(X'X) X'$$

can be used as a projection matrix. The hat matrix diagonal variable contains the diagonal elements of the hat matrix

$$h_i = x_i \text{ inv}(X'X) x_i'$$

Overrides: *gmisclib.gpk_ols.ols_base.hat*

rank(*self*)

residual(*self*)

Overrides: *gmisclib.gpk_ols.ols_base.residual*

set_y(*self*, *y*, *copy*=True)

Overrides: *gmisclib.gpk_ols.ols_base.set_y*

sv(*self*)

variance_about_fit(*self*)

Returns the estimator of the standard deviation of the data about the fit.

Return Value

`numpy.ndarray` with shape=(q). Each entry corresponds to one of the q sets of equations that are being fit.

x(*self*, *y*=None, *copy*=True)

x_variances(*self*)

Estimated standard deviations of the solution. This is the diagonal of the solution covariance matrix.

<code>y(self, copy=True)</code> Overrides: <code>gmisclib.gpk_ols.base.y</code>
--

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

60.5.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

60.5.3 Class Variables

Name	Description
HTMIN	Value: 3.0
STMIN	Value: 1.5

61 Module gmisclib.gpk_writer

61.1 Variables

Name	Description
<code>--package--</code>	Value: 'gmisclib'

61.2 Class writer

object └─ **gmisclib.gpk_writer.writer**

Known Subclasses: gmisclib.avio.writer, gmisclib.fiatio.writer, gmisclib.gpk_writer.null_writer, select_fiat_entries.writer

61.2.1 Methods

comments(*self*, *comments*)

Add comments to the data file. Comments can appear anywhere.

comment(*self*, *comment*)

header(*self*, *k*, *v*)

headers(*self*, *h*)

__init__(*self*, *fd*)

x.__init__(...) initializes *x*; see *x.__class__.__doc__* for signature

Overrides: object.__init__ extit(inherited documentation)

data(*self*, *dataset*)

Write a series of lines to the output file.

extend(*self*, *d*)

append(*self*, *d*)

datum(*self*, *data_item*)

flush(*self*)

close(*self*)

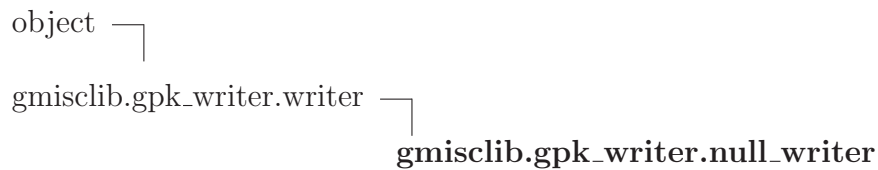
Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

61.2.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

61.3 Class `null_writer`



61.3.1 Methods

comments(*self*, *comments*)

Add comments to the data file. Comments can appear anywhere.

Overrides: `gmisclib.gpk_writer.writer.comments` `exitit`(inherited documentation)

comment(*self*, *comment*)

Overrides: `gmisclib.gpk_writer.writer.comment`

header(*self*, *k*, *v*)

Overrides: `gmisclib.gpk_writer.writer.header`

headers(*self*, *h*)

Overrides: *gmisclib.gpk_writer.writer.headers*

__init__(*self*)

x.__init__(...) initializes *x*; see *x.__class__.__doc__* for signature

Overrides: *object.__init__* *exitit*(inherited documentation)

data(*self*, *dataset*)

Write a series of lines to the output file.

Overrides: *gmisclib.gpk_writer.writer.data* *exitit*(inherited documentation)

datum(*self*, *data_item*)

Overrides: *gmisclib.gpk_writer.writer.datum*

__del__(*self*)

flush(*self*)

Overrides: *gmisclib.gpk_writer.writer.flush*

close(*self*)

Overrides: *gmisclib.gpk_writer.writer.close*

append(*self*, *d*)

extend(*self*, *d*)

Inherited from object

__delattr__(), *__format__*(), *__getattribute__*(), *__hash__*(), *__new__*(), *__reduce__*(), *__reduce_ex__*(),
__repr__(), *__setattr__*(), *__sizeof__*(), *__str__*(), *__subclasshook__*()

61.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<i>__class__</i>	

62 Module *gmisclib.gpkequations*

62.1 Functions

range_of_indices(*l*)

This finds the highest and lowest indices for a list of equations.

bandwidth(*l*)

build_matrices(*l*, *n*=None, *maxbw*=None)

Takes a list of *eqn_c* classes, and builds the matrices to be solved.

test_eqn()

add_something(*a*, *thing*, *scale*)

Add a band-diagonal part to matrix *a*. We assume *a* has shared-reference semantics, like the Numeric module. This will **not** work for *sbd_array*, which has copy semantics for *__getslice__*.

add_slope_curv_droop(*a*, *sls*, *cs*, *droop*)

Adds a slope and curvature part to the matrix. Returns the bandwidth of the part it added.

full_matrices(*l*, *n*, *sls*, *cs*, *droop*)

solve_eqns(*a*, *b*)

test_solve()

test_solve2()

test_bw()

test_as()

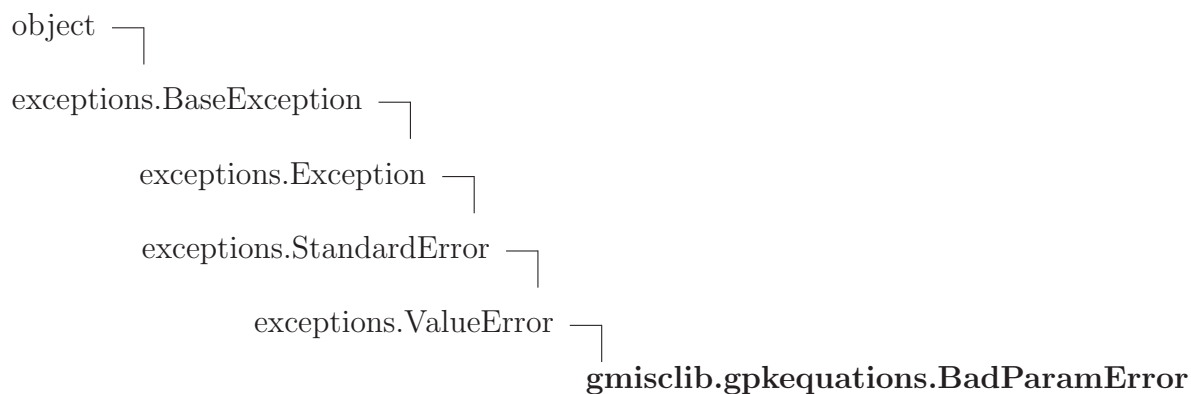
Won't work for *sbd_array*.

test()

62.2 Variables

Name	Description
<code>--package--</code>	Value: <code>'gmisclib'</code>

62.3 Class *BadParamError*



62.3.1 Methods

```

__init__(self, s)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature
Overrides: object.__init__ extit(inherited documentation)

```

Inherited from `exceptions.ValueError`

```
__new__()
```

Inherited from `exceptions.BaseException`

```

__delattr__(), __getattr__(), __getitem__(), __getslice__(), __reduce__(), __repr__(),
__setattr__(), __setstate__(), __str__(), __unicode__()

```

Inherited from `object`

```
__format__(), __hash__(), __reduce_ex__(), __sizeof__(), __subclasshook__()
```

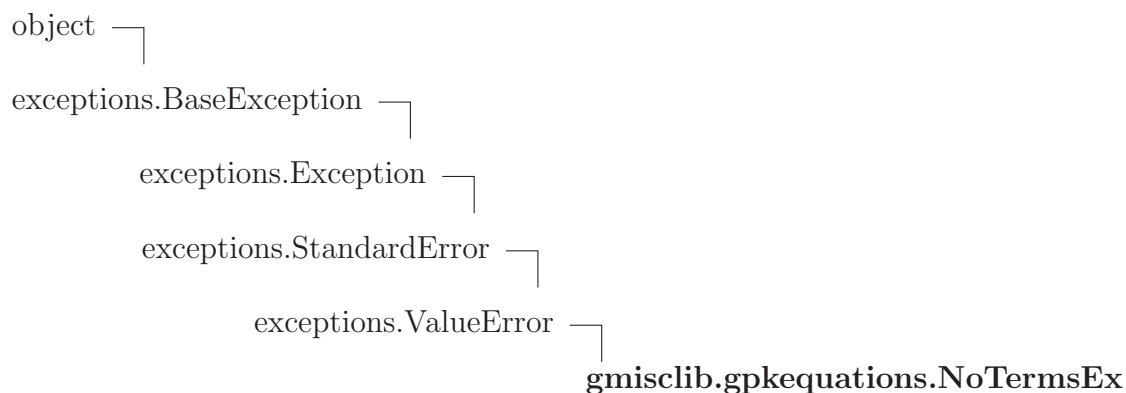
62.3.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	

continued on next page

Name	Description
args, message	
<i>Inherited from object</i>	
__class__	

62.4 Class NoTermsEx



62.4.1 Methods

```

__init__(self, s='Empty Equation')
x.__init__(...) initializes x; see x.__class__.__doc__ for signature
Overrides: object.__init__ extit(inherited documentation)

```

Inherited from exceptions.ValueError

```
__new__()
```

Inherited from exceptions.BaseException

```

__delattr__(), __getattr__(), __getitem__(), __getslice__(), __reduce__(), __repr__(),
__setattr__(), __setstate__(), __str__(), __unicode__()

```

Inherited from object

```
__format__(), __hash__(), __reduce_ex__(), __sizeof__(), __subclasshook__()
```

62.4.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	

continued on next page

Name	Description
args, message	
<i>Inherited from object</i>	
__class__	

62.5 Class eqn_c

62.5.1 Methods

index_ends(*self*)

__init__(*self, terms, sum, wt, index_ctr*)

Takes an equation in the form: [(index, coefficient), ...] = sum and stores it away into a class.

__str__(*self*)

substitute(*self, known, zero=0*)

Modifies an equation by substituting in known values. Argument 'known' is an array of values.

__nonzero__(*self*)

63 Module *gmisclib.gpkmisc*

63.1 Functions

median (x)

avg (x)

median_ad (x)

Median absolute deviation

mad (x)

Median absolute deviation

mean_ad (x)

geo_mean (* d)

Return Value

Geometric mean of its arguments.

(*type=float*)

Raises

ValueError if any argument is negative.

entropy (x)

Compute the entropy of a list of probabilities.

resample (d)

Bootstrap resampling. Call this many times: each one returns a random resampling.

jackknife (d)

Jackknife resampling. Call this once. It returns a list of deleted lists.

Student_t_dens (x, n)

From p.337 Statistical Theory by Bernard Lindgren.
--

log_Student_t_dens(*x*, *n*)

From p.337 Statistical Theory by Bernard Lindgren.

log_factorial(*n*)

log_Combinations(*n*, *m*)

ComplexMedian(*P*)

P is a list of complex numbers. This algorithm works by repeatedly stripping off the convex hull of the points.

testCM()

thr_iter_read(*fd*)

Read the contents of a file as an iterator. The read is two-threaded, so that one thread can be waiting on disk I/O while the other thread is processing the results.

makedirs(*fname*, *mode*=509)

This makes the specified directory, including all necessary directories above it. It is like `os.makedirs()`, except that if the directory already exists it does not raise an exception.

Parameters

fname: Name of the directory to create.

(*type*=*str*)

mode: Linux file permissions for any directories it needs to create.

(*type*=*int*)

Raises

OSError If it cannot create a part of the directory chain.

Note: If the directory already exists, it does not force it to have the specified mode.

shuffle_Nrep(*y*, *n*=1, *compare*=None)

Shuffle a list, *y*, so that no item occurs more than *n* times in a row. Equality is determined by the comparison function *compare* returning zero.

testSNR()

open_nowipe(*nm1*, *nm2*, *mode*='w')

Open a file (typically for writing) but make sure that the file doesn't already exist. The name is constructed from *nm1*, a sequence number, and *nm2*. The sequence number gets incremented until a name is found that doesn't exist. This works by creating a directory as a lock file; it should be safe across NFS.

Parameters

nm1: the part of the name to the left of the sequence number

(*type=**str*)

nm2: the part of the name to the right of the sequence number

Typically, *nm2* is a suffix like ".wav". This may not contain a slash.

(*type=**str*)

mode: The way to open the file – passed to `open()`.

(*type=**str @rtype file*)

Return Value

The opened `file` object. (Its name can be gotten from the **name** attribute.)

Note: The directory containing **nm1** must exist and be writeable.

dropfront(*prefix*, *s*)

Drops the prefix from the string and raises an exception if it is not there to be dropped.

open_compressed(*fn*)

gamma*ln*(*x*)

a_factor(*n*)

Finds the smallest prime factor of a number.

primes()

This is a generator that produces an infinite list of primes.

factor(*n*)

Factor a number into a list of prime factors, in increasing order.

Parameters

n: input number
(*type=int*)

Return Value

prime factors
(*type=list*)

test_primes()**gcd**(*a, b*)

Greatest common factor/denominator.

Parameters

a: (*type=int*)
b: (*type=int*)

Return Value

the greatest common factor of a and b.
(*type=int*)

find_in_PATH(*progrname*)

Search PATH to find where a program resides.

Parameters

progrname: the program to look for.
(*type=str*)

Return Value

the full path name.
(*type=str*)

get_mtime(*fn*)

Paired with `need_to_recompute()`. These implement something like make, where we figure out if we need to compute things based on the age of files. This is used to get the age of the pre-requisites.

Parameters

fn: a filename or a file
(type=str or file)

Return Value

None (if the file doesn't exist) or its modification time.
(type=None or float)

prereq_mtime(tlist*)****need_to_recompute(*fn*, *lazytime*, *size*=-1)**

Paired with `get_mtime()`. These implement something like make, where we figure out if we need to compute things.

Parameters

fn: a filename or a file
(type=str or file)

lazytime: a time (as obtained from `ST_MTIME` in `os.stat()`). If the file modification time of **fn** is older than **lazytime**, recompute.
(type=None or float)

size: recompute the file if it is smaller than **size**. Normally, this is used to recompute on empty output files by setting **size**=0.
(type=int)

Return Value

True if **fn** needs to be recomputed or if it doesn't exist.
(type=bool)

truncate(*s*, *maxlen*)**erf(*x*)**

$\text{erf}(x) = (2/\sqrt{\pi}) * \int_0^x \exp(-t^2) dt$

asinh(*x*)

Inverse hyperbolic sine.

chooseP(*x*, *p*)

Sample from a list with specified probabilities.

Parameters**x**: a list of things from which to sample*(type=list(something))***p**: a list of probabilities for sampling the corresponding item of **x**.*(type=list(float))***Return Value**a sample from **x***(type=whatever is inside x)***Raises****AssertionError** Will (sometimes) detect negative probabilities, or probabilities that sum to something other than one.

63.2 Variables

Name	Description
<code>__package__</code>	Value: <code>'gmisclib'</code>

63.3 Class `threaded_readable_file`

object — `gmisclib.gpkmisc.threaded_readable_file`

63.3.1 Methods

__init__(*self*, *fd*)*x.__init__*(...) initializes *x*; see *x.__class__.__doc__* for signatureOverrides: *object.__init__* *exitit*(inherited documentation)**readline**(*self*)

<code>readlines(self)</code>

<code>read_iter(self)</code>

<code>__iter__(self)</code>

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

63.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

63.3.3 Class Variables

Name	Description
<code>QSIZE</code>	Value: 100

63.4 Class `dir_lock`

object —
 `gmisclib.gpkmisc.dir_lock`

63.4.1 Methods

<code>__init__(self, lockname)</code>

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` `exitit`(inherited documentation)

<code>__enter__(self)</code>

<code>__exit__(self, exc_type, exc_value, traceback)</code>

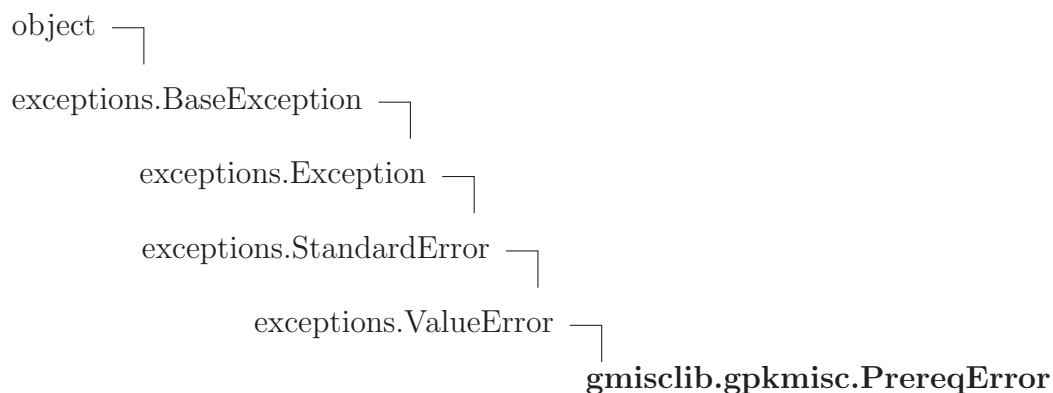
Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

63.4.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

63.5 Class *PrereqError*



63.5.1 Methods

<code>__init__(self, *s)</code> <code>x.__init__(...)</code> initializes x; see <code>x.__class__.__doc__</code> for signature Overrides: <code>object.__init__</code> extit(inherited documentation)
<code>repl(self, x)</code>

Inherited from `exceptions.ValueError`

`__new__()`

Inherited from `exceptions.BaseException`

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

Inherited from `object`

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

63.5.2 Properties

Name	Description
	<i>Inherited from exceptions.BaseException</i>
	args, message
	<i>Inherited from object</i>
<code>__class__</code>	

64 Module *gmisclib.hilbert_xform*

64.1 Functions

nextpow2 (<i>i</i>)

hilbert (<i>x</i> , <i>cutoff</i>)

This is a Hilbert transform when cutoff=0.
--

64.2 Variables

Name	Description
--package--	Value: 'gmisclib'

65 Module `gmisclib.kl_dist`

Suppose there is a random variable with true distribution p . Then (as we will see) we could represent that random variable with a code that has average length $H(p)$. However, due to incomplete information we do not know p ; instead we assume that the distribution of the random variable is q . Then (as we will see) the code would need more bits to represent the random variable. The difference in the number of bits is denoted as $D(p|q)$. The quantity $D(p|q)$ comes up often enough that it has a name: it is known as the relative entropy:

The relative entropy or Kullback-Leibler distance between two probability mass functions $p(x)$ and $q(x)$ is defined as
 $D(p||q) = \text{Sum}\{x \text{ in } X\} p(x) \log(p(x)/q(x))$

Note that this is not symmetric, and the q (the second argument) appears only in the denominator.

65.1 Functions

P (x)

multinomial_logp (x, cF)

multinomial_fixer (x, c)

kl_nonzero_probs (p, q)

Kullback-Liebler distance between two normalized, nonzero probability distributions.
--

kl_nonzero_prob_m (p, q)

kl_dist_vec ($p, q, N=\text{None}, Fp=1.0, Fq=1.0, Clip=0.01$)

Relative entropy or Kullback-Liebler distance between two frequency distributions p and q . Here, we assume that both p and q are counts derived from multinomial distributed data; they are not normalized to one.

tr_from_obs (x)

Given a matrix of $P(i \text{ and } j)$ as $x[i,j]$, we compute $P(j \text{ given } i)$ and return it as $y[i,j]$. The result is a transition probability matrix where the first index is the input state, and the second index marks the result state.

estimate_tr_probs (<i>counts</i> , <i>N</i> , <i>F</i> =1.0)
--

solve_for_pi (<i>p</i>)

Given a transition probability matrix <i>p</i> , where the first index is the initial state and the second index is the resultant state, compute the steady-state probability distribution, assuming a Markov process.
--

kl_nonzero_tr_probs (<i>pp</i> , <i>qq</i>)
--

KL distance, given a matrix of nonzero transition probabilities. Each matrix indexes states as <i>pp</i> [from,to], and contains $P(\text{to given from})$ as a conditional probability, where for any from, the Sum over to(<i>pp</i> [from,to]) = 1.

kl_nonzero_tr_prob_m (<i>pp</i> , <i>qq</i>)

cross (<i>a</i> , <i>b</i>)

kldist_Markov (<i>p</i> , <i>q</i> , <i>N</i> =None)
--

Kullback-Liebler distance between two matrices of bigram counts.
--

kldist_Markov_m (<i>p</i> , <i>q</i> , <i>N</i> =None)
--

Kullback-Liebler distance between two matrices of bigram counts.
--

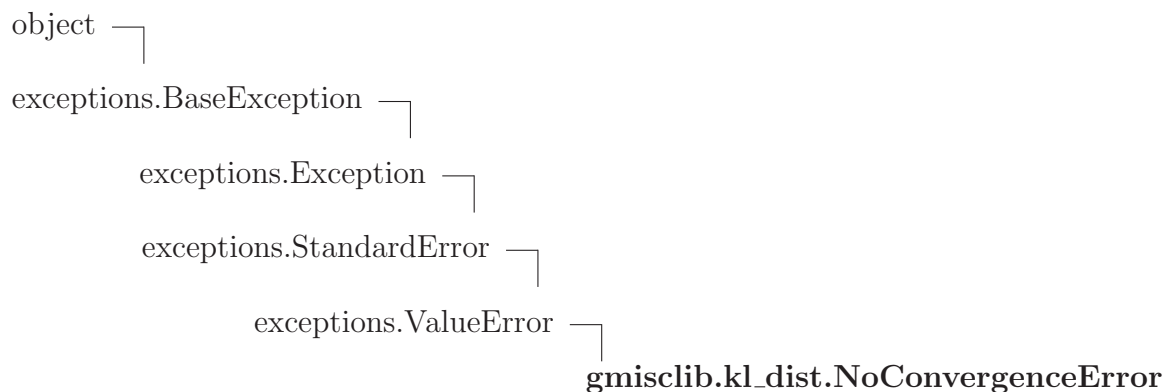
kldist_Markov_mm (* <i>p</i>)

List of Kullback-Liebler distances between all combinations of pairs of matrices of bigram counts. It returns a list of matrices of all the distances. Each item on the list is a sample of the distance histogram.

65.2 Variables

Name	Description
<code>__package__</code>	Value: 'gmisclib'

65.3 Class NoConvergenceError



65.3.1 Methods

`__init__(self, s)`
`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature
 Overrides: `object.__init__` extit(inherited documentation)

Inherited from `exceptions.ValueError`

`__new__()`

Inherited from `exceptions.BaseException`

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

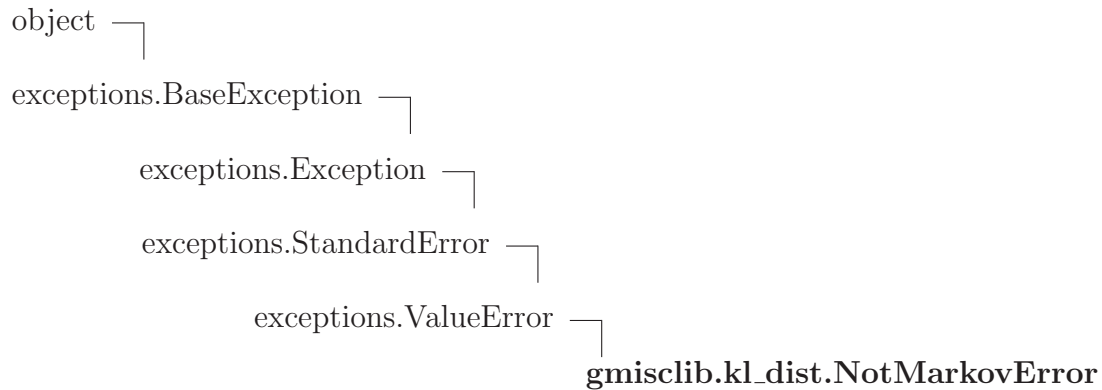
Inherited from `object`

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

65.3.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
<code>args</code> , <code>message</code>	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

65.4 Class `NotMarkovError`



65.4.1 Methods

`__init__`(*self*, *s*)
x.**`__init__`**(...) initializes *x*; see *x*.**`__class__`**.**`__doc__`** for signature
 Overrides: `object.__init__` extit(inherited documentation)

Inherited from `exceptions.ValueError`

`__new__`()

Inherited from `exceptions.BaseException`

`__delattr__`(), **`__getattr__`**(), **`__getitem__`**(), **`__getslice__`**(), **`__reduce__`**(), **`__repr__`**(),
`__setattr__`(), **`__setstate__`**(), **`__str__`**(), **`__unicode__`**()

Inherited from `object`

`__format__`(), **`__hash__`**(), **`__reduce_ex__`**(), **`__sizeof__`**(), **`__subclasshook__`**()

65.4.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
args, message	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

66 Module `gmisclib.lapack_dgesv`

66.1 Functions

<code>dgesv(<i>a</i>, <i>b</i>)</code>
Solve $a \cdot x = b$.

67 Module `gmisclib.lapack_dsyevd`

67.1 Functions

<code>dsyevd(<i>a</i>, <i>jobz</i>='V', <i>uplo</i>='U')</code>
Eigenvalues and eigenvectors of a real symmetric matrix.

68 Module `gmisclib.load_mod`

This module has functions that help you dynamically import modules.

68.1 Functions

`split_name(name)`

Split a name in the form `a/b.c` into `a`, `b`, `c`, where `a` is a search path, `b` is a module (package) name, and `c` is a name in the module.

`load(name, path)`

Load a module from the specified list of paths, or, failing that, look in `sys.path` then for a builtin. It returns the module, but does not import it. If `path` is `None`, only look in `sys.path` and builtins. If `path` is an array containing `None`, replace the `None` with `sys.path`.

This version gives you detailed control over the search path.

`load_mod(name, path)`

Load a module from the specified list of paths, or, failing that, look in `sys.path` then for a builtin. It returns the module, but does not import it. If `path` is `None`, only look in `sys.path` and builtins. If `path` is an array containing `None`, replace the `None` with `sys.path`.

This version gives you detailed control over the search path.

`load_inc_path(name, path)`

Load a module from the specified list of paths, or, failing that, look in `sys.path` then for a builtin. It returns the module, but does not import it. If `path` is `None`, only look in `sys.path` and builtins. If `path` is an array containing `None`, replace the `None` with `sys.path`.

This version gives you detailed control over the search path.

load_mod_inc_path(*name*, *path*)

Load a module from the specified list of paths, or, failing that, look in `sys.path` then for a builtin. It returns the module, but does not import it. If *path* is `None`, only look in `sys.path` and builtins. If *path* is an array containing `None`, replace the `None` with `sys.path`.

This version gives you detailed control over the search path.

load_named(*name*, *use_sys_path*=`True`)

Load a module. If the module name is in the form `a/b`, it looks in directory `"a"` first. If *use_sys_path* is `true`, it searches the entire Python path

It returns the module, but does not import it. This version handles importing packages and functions nicely, but with less control over the search path.

Usage:

- `load_named_module('/dir/my_module')`, or
- `load_named_module('foo/my_module')`, or
- `load_named_module('foo/my_module.submodule.function')`, or
- various combinations.

load_named_module(*name*, *use_sys_path*=`True`)

Load a module. If the module name is in the form `a/b`, it looks in directory `"a"` first. If *use_sys_path* is `true`, it searches the entire Python path

It returns the module, but does not import it. This version handles importing packages and functions nicely, but with less control over the search path.

Usage:

- `load_named_module('/dir/my_module')`, or
- `load_named_module('foo/my_module')`, or
- `load_named_module('foo/my_module.submodule.function')`, or
- various combinations.

load_named_fcn(*name*, *use_sys_path*=True)

Load a module. If the module name is in the form a/b, it looks in directory "a" first. If *use_sys_path* is true, it searches the entire Python path

It returns the module, but does not import it. This version handles importing packages and functions nicely, but with less control over the search path.

Usage:

- `load_named_module('/dir/my_module')`, or
- `load_named_module('foo/my_module')`, or
- `load_named_module('foo/my_module.submodule.function')`, or
- various combinations.

load_fcn(*name*, *use_sys_path*=True)

Load a module. If the module name is in the form a/b, it looks in directory "a" first. If *use_sys_path* is true, it searches the entire Python path

It returns the module, but does not import it. This version handles importing packages and functions nicely, but with less control over the search path.

Usage:

- `load_named_module('/dir/my_module')`, or
- `load_named_module('foo/my_module')`, or
- `load_named_module('foo/my_module.submodule.function')`, or
- various combinations.

68.2 Variables

Name	Description
<code>--package--</code>	Value: 'gmisclib'

69 Module *gmisclib.lreg_fill*

69.1 Functions

<code>fill(<i>f</i>, <i>wt</i>, <i>s</i>)</code>
--

69.2 Variables

Name	Description
LREG	Value: <code>'/home/gpk/misctts/lreg_fit2'</code>
<code>--package--</code>	Value: <code>'gmisclib'</code>

70 Module **gmisclib.makemake**

This module is designed to build makefiles.

70.1 Functions

maketemp (<i>dir</i> =' /tmp', <i>prefix</i> ='maketemp', <i>suffix</i> ='')
--

setlog (<i>f</i>)

log (* <i>s</i>)

Log some strings, one per line.

quote (<i>s</i>)

var (<i>k</i> , <i>v</i>)

Pass a variable to make.

rule (<i>a</i> , * <i>b</i>)

Writes a rule into a makefile. All lines except the first are indented.

blank ()

set_debug ()

finish ()

date ()

path_to (<i>s</i>)

set_make_prog (* <i>s</i>)

`ncpu()`

Return Value

a string representation of the integer number of cores that the computer has.

(*type=*str !Not an integer!)

`read(fn)`

70.2 Variables

Name	Description
<code>makeflags</code>	Value: []
<code>makefile</code>	Value: None
<code>DB_format</code>	Value: 'fiatio'
<code>__package__</code>	Value: 'gmisclib'

70.3 Class `FileNotFound`



70.3.1 Methods

`__init__(self, *s)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` extit(inherited documentation)

Inherited from `exceptions.Exception`

`__new__()`

Inherited from `exceptions.BaseException`

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

Inherited from object

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

70.3.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
args, message	
<i>Inherited from object</i>	
__class__	

71 Module `gmisclib.matrix_arrange`

Take a covariance-like matrix, and re-order it to be close to a diagonal matrix. We calculate a map that renames the indices (the same map for both left and right), to bring large off-diagonal elements close to the diagonal.

71.1 Functions

<code>diagonalize($a\theta$)</code>
--

<code>map_array(a, m)</code>

<code>test()</code>

71.2 Variables

Name	Description
<code>--package--</code>	Value: <code>'gmisclib'</code>

72 Module `gmisclib.matrix_arrange_entropy`

This rotates a matrix by multiplying with a unitary matrix so that the resulting elements are either nearly zero or relatively large. It minimizes sum of $|x_{ij}| \log(|x_{ij}|)$, i.e. the entropy (sortof).

72.1 Functions

<code>make_min_entropy</code> (<i>x</i> , <i>extra_entropy</i> =0.0)
--

<i>extra_entropy</i> is a vector that helps choose what to optimize.
--

<code>test2</code> ()

<code>test3</code> ()

72.2 Variables

Name	Description
<code>--package--</code>	Value: <code>'gmisclib'</code>

73 Module `gmisc.lib.matrix_rearrange_labels`

This module helps you plot confusion matrices or similar images where the axis labels do not have a natural order. So, the probability of confusing two phonemes is a perfect example: phonemes do not naturally fall onto a 1-dimensional sequence, so one is free to put them in any order one likes. Given that, one might as well put them into an order that reveals something interesting about the probabilities.

All the functions ending in "2" work on rectangular arrays. All the functions without "2" work only on square arrays, and they assume that both axes remain in the same order.

73.1 Functions

<code>symm_swap_toward_minimal_fom</code> <i>(<code>m</code>, <code>fom</code>, <code>maxtries</code>)</i>
Used internally. Finds an ordering of the labels that minimizes whatever <code>fom</code> you supply. Both axes will have the same order.

<code>swap_toward_minimal_fom</code> <i>(<code>m</code>, <code>fom</code>, <code>maxtries</code>)</i>
Used internally. Finds an ordering of the labels that minimizes whatever <code>fom</code> you supply. The ordering of the two axes may be different.

swap_toward_diag(*m*, *lbls*, *maxtries*=None, *sign*=1)

Swap the rows and columns of a matrix to bring it closer to a diagonal matrix: i.e. entries with large absolute values on the main diagonal and small entries away from the main diagonal. Rows and columns are swapped together, so that the ordering of rows will match the ordering of columns.

Parameters

- m:** a 2-dimensional array
(*type*=`numpy.ndarray`)
- lbls:** a list of arbitrary labels for each row or column.
(*type*=`list(anything)`)
- maxtries:** how hard to work at optimizing the matrix layout? None gives a resonable default value. Appropriate values are a few times the number of elements in the matrix.
(*type*=`int` or `None`)
- sign:** (default=1) if **sign**=-1, do the opposite: put the small absolute values on the main diagonal.
(*type*=`int`)

Return Value

A tuple containing the swapped matrix and a swapped list of values.
(*type*=`tuple(numpy.ndarray, list(something))`)

swap_toward_positive(*m*, *lbls*, *maxtries*=None, *sign*=1)

Swap the rows and columns of a matrix to put the most positive values on the main diagonal. Rows and columns are swapped together, so that the ordering of rows will match the ordering of columns.

Parameters

- m:** a 2-dimensional array
(*type*=*numpy.ndarray*)
- lbls:** a list of arbitrary labels for each row or column.
(*type*=*list(anything)*)
- maxtries:** how hard to work at optimizing the matrix layout? None gives a resonable default value. Appropriate values are a few times the number of elements in the matrix.
(*type*=*int* or *None*)
- sign:** (default=1) if **sign**=-1, do the opposite: put the negative entries on the main diagonal.
(*type*=*int*)

Return Value

A tuple containing the swapped matrix and a swapped list of labels.
(*type*=*tuple(numpy.ndarray, list(something))*)

swap_toward_diag2(*m*, *lbl1*, *lbl2*, *maxtries*=None)

Swap the rows and columns of a matrix to make it roughly diagonal: i.e. large entries on the main diagonal and small entries away from the main diagonal. Note that this will work even if the matrix is not square.

Parameters

- m:** a 2-dimensional array
(*type*=`numpy.ndarray`)
- lbl1:** a list of arbitrary labels for the first index of the matrix **m**.
(*type*=`list(anything)`)
- lbl2:** a list of arbitrary labels for the second index of the matrix **m**.
(*type*=`list(anything)`)
- maxtries:** how hard to work at optimizing the matrix layout? None gives a resonable default value. Appropriate values are a few times the number of elements in the matrix.
(*type*=`int` or `None`)

Return Value

A tuple containing the swapped matrix and the two swapped lists of labels, one for the first and one for the second axis.
(*type*=`tuple(numpy.ndarray, list(something), list(something))`)

pos_near_diag2(*m*, *lbl1*, *lbl2*, *maxtries*=None)

neg_near_diag2(*m*, *lbl1*, *lbl2*, *maxtries*=None)

swap_toward_blocks2(*m*, *lbl1*, *lbl2*, *maxtries*=None)

Swap rows and columns of a matrix to bring it closer to a block form, where similar values occur together in blocks.

Parameters

- m:** a 2-dimensional array
(*type*=*numpy.ndarray*)
- lbl1:** a list of arbitrary labels for the first axis of m
(*type*=*list(anything)*)
- lbl2:** a list of arbitrary labels for the second axis of m
(*type*=*list(anything)*)
- maxtries:** how hard to work at optimizing the matrix layout? None gives a resonable default value. Appropriate values are a few times the number of elements in the matrix.
(*type*=*int* or *None*)

Return Value

A tuple containing the swapped matrix and the two swapped lists of values, one for the first and one for the second axis.

(*type*=*tuple(numpy.ndarray, list(something), list(something))*)

swap_toward_blocks(*m*, *lbls*, *maxtries*=None)

Swap rows and columns of a matrix to bring it closer to a block form, where similar values occur together in blocks. It minimizes the sum of changes when proceeding along a row or column.

Parameters

- m:** a 2-dimensional array
(*type*=*numpy.ndarray*)
- lbls:** a list of arbitrary labels for the first axis of m
(*type*=*list(anything)*)
- maxtries:** how hard to work at optimizing the matrix layout? None gives a resonable default value. Appropriate values are a few times the number of elements in the matrix.
(*type*=*int* or *None*)

Return Value

A tuple containing the swapped matrix and a swapped lists of values.

(*type*=*tuple(numpy.ndarray, list(something))*)

test1()

Two test cases on 2x2 matrices.

test()

73.2 Variables

Name	Description
<code>__package__</code>	Value: 'gmisclib'

73.3 Class diagfom

object └─
 gmisclib.matrix_rearrange_labels.diagfom

A class that defines a figure-of-merit for a matrix. It tries to put the largest entries on the diagonal. Instances of this class are passed to `symm_swap_toward_minimal_fom`.

73.3.1 Methods

<code>__init__(self, n, sign=1)</code>

<code>x.__init__(...)</code> initializes x; see <code>x.__class__.__doc__</code> for signature
--

Parameters

- sign:** • If `sign=1`, the f-o-m returned by `eval` will be minimal when the most positive matrix elements are on the diagonal.
- If `sign=-1`, you'll get minimal fom with the the most positive elements off diagonal.

(*type=int*)

n: dimension of matrix.

(*type=int*)

Overrides: <code>object.__init__</code>

```
eval(self, swv, m)
```

Evaluate the figure of merit for a matrix m with the specified swap vector.

Parameters

m: matrix

swv: swap vector

Return Value

figure of merit

(type=int or float (according to the type of m).)

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

73.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

73.4 Class diagfom2

```
object └─ gmisclib.matrix_rearrange_labels.diagfom2
```

This tries to make your matrix approximately diagonal for rectangular matrices.

73.4.1 Methods

```
__init__(self, n1, n2)
```

x.`__init__`(...) initializes x; see x.`__class__`.`__doc__` for signature

Overrides: object.`__init__` extit(inherited documentation)

```
eval(self, sw1, sw2, m)
```

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

73.4.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

73.5 Class *blockfom*

object └─ **`gmisc.lib.matrix_rearrange_labels.blockfom`**

This tries to make your matrix into a block form. The idea is to minimize the sum of entry-to-entry differences along a row (same with columns).

73.5.1 Methods

<code>__init__(self, n1, n2)</code>
<code>x.__init__(...)</code> initializes <code>x</code> ; see <code>x.__class__.__doc__</code> for signature
Overrides: <code>object.__init__</code> extit(inherited documentation)
<code>eval(self, sw1, sw2, m)</code>

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

73.5.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

74 Module *gmisclib.mcmc*

Adaptive Markov-Chain Monte-Carlo algorithm. This can be used to generate samples from a probability distribution, or also as a simulated annealing algorithm for maximization. This can be imported and its functions and classes can be used.

The central interfaces are the `mcmc.BootStepper` class, and within it, the `step()` method is used iteratively to take a Markov step.

The algorithm is described in Kochanski and Rosner 2010, and earlier versions have been used in ZZZ. It evolved originally from `amoeba_anneal` (in Numerical Recipes, Press et al.). The essential feature is that it keeps a large archive of previous positions (possibly many times more than N of them). It samples two positions from the archive and subtracts them to generate candidate steps. This has the nice property that when sampling from a multivariate Gaussian distribution, the candidate steps match the distribution nicely.

It can be operated in two modes (or set to automatically switch). One is optimization mode where it heads for the maximum of the probability distribution. The other mode is sampling mode, where it asymptotically follows a Markov sampling procedure and has the proper statistical properties.

74.1 Functions

start_is_list_a (<i>start</i>)

Is the argument a sequence of numpy arrays?

start_is_list_p (<i>start</i>)

Is the argument a sequence of position_base objects?
--

make_list_of_positions (<i>x</i> , <i>PositionClass</i> , <i>problem_def</i>)
--

Turn almost anything into a list of position_base objects. You can hand it a sequence of numpy vectors or a single 1-dimensional numpy vector; a sequence of position_base objects or a single 1-dimensional position_base object.
--

Precondition: This depends on <code>PositionClass</code> being callable as <code>PositionClass(vector_of_doubles, problem_definition)</code> .

bootstepper (<i>logp</i> , <i>x</i> , <i>v</i> , <i>c=None</i> , <i>sort='auto'</i> , <i>fixer=None</i> , <i>repeatable=True</i>)
--

This is (essentially) another interface to the class constructor. It's really there for backwards compatibility.
--

```
boot2stepper(logp, x, v, c=None, sort='auto', fixer=None,
repeatable=True)
```

This is (essentially) another interface to the class constructor. It's really there for backwards compatibility.

```
test(stepper)
```

```
diag_variance(start)
```

Hand this a list of vectors and it will compute the variance of each component, then return a diagonal covariance matrix.

74.2 Variables

Name	Description
Debug	Value: 0
MEMORYS_WORTH_OF_PARAMETERS	Value: 1000000000.0
--package--	Value: 'gmisclib'

74.3 Class `problem_definition`

object └─ **gmisclib.mcmc.problem_definition**

This class implements the problem to be solved. It's overall function in life is to compute the probability that a given parameter vector is acceptable. `Mcmc.py` then uses that to run a Markov-Chain Monte-Carlo sampling. You may want to derive a class from this and override all the functions defined here, or implement your own class with the same functions. (Of course, in that case, it can contain extra functions also.)

For instance, it can be a good idea to define a function "model" that computes the model that you are fitting to data (if that is your plan). Then `logp()` can be something like `-numpy.sum((self.model()-self.data)**2)`. Also, it can be good to define a "guess" function that computes a reasonable initial guess to the parameters, somehow.

74.3.1 Methods

`__init__(self, logp_fcn, c, fixer=None)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` `exitit`(inherited documentation)

`logp(self, x)`

Compute the log of the probability density at `x`.

Parameters

`x`: a parameter vector

(*type=*`numpy.ndarray`)

Return Value

The log of the probability that the model assigns to parameters `x`.

(*type=*`float`)

Raises

NotGoodPosition This exception is used to indicate that the position `x` is not valid. This is equivalent to returning an extremely negative `logp`.

fixer(*self*, *x*)

This is called on each candidate position vector. Generally, it is used to restrict the possible solution space by folding position vectors that escape outside the solution space back into the solution space. It can also allow for symmetries in equations.

Formally, it defines a convex region. All vectors outside the region are mapped into the region, but the mapping must be continuous at the boundary. (More precisely, $\text{logp}(\text{fixer}(x))$ must be continuous everywhere that $\text{logp}(x)$ is continuous, including the boundary.) For instance, mapping $x[0]$ into $\text{abs}(x[0])$ defines a convex region (the positive half-space), and the mapping is continuous near $x[0]=0$.

Additionally, it may re-normalize parameters at will subject to the restriction that $\text{logp}(\text{fixer}(x)) = \text{logp}(x)$. For instance, it can implement a constraint that $\text{sum}(x) = 0$ by mapping x into $x - \text{average}(x)$, so long as the value of $\text{logp}()$ is unaffected by that substitution. other folds can sometimes lead to problems.

Parameters

x: a parameter vector
(*type=*`numpy.ndarray`)

Return Value

a (possibly modified) parameter vector.
(*type=*`numpy.ndarray`)

Raises

NotGoodPosition This exception is used to indicate that the position *x* is not valid. Fixer has the option of either mapping invalid parameter vectors into valid ones or raising this exception.

Attention: Within a convex region (presumably one that contains the optimal *x*), *fixer* must **not** change the value of $\text{logp}()$: $\text{logp}(\text{fixer}(x)) = \text{logp}(x)$.

log(*self*, *p*, *i*)

Some code calls this function every iteration to log the current state of the MCMC process.

Parameters

p: the current parameter vector, and
i: an integer iteration counter.

Return Value

nothing.

guess(*self*)

Some code calls this function at the beginning of the MCMC process to seed the iteration.

Return Value

a list of guess vectors; each guess vector is suitable for passing to `self.fixer()` or `self.logp()` as `x`.

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

74.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

74.4 Class `position_base`

```

object └─
          gmislib.mcmc.position_base

```

Known Subclasses: `gmislib.mcmc.position_history_dependent`, `gmislib.mcmc.position_nonrepeatable`, `gmislib.mcmc.position_repeatable`

This class is used internally in the MCMC sampling process to represent a position. It stores a parameter vector, a reference to the problem definition, and (optionally) caches computed values of $\log(P)$.

74.4.1 Methods**`__init__(self, x, problem_def)`**

You need this function and this signature if you are going to pass it to `make_list_of_positions()`, but not necessarily otherwise.

Overrides: `object.__init__`

`logp(self)`

Compute the log of the probability for the position.

logp_nocompute(*self*)

Shows a recent logp value. It does not compute a value unless none has ever been computed.

new(*self*, *shift*, *logp*=None)

Returns a new position, shifted by the specified amount.

Parameters

shift: How much of a move to make to the new position.

(*type*=`numpy.ndarray`)

logp: (optional) If this is supplied, is is used to set the log(P) value for the newly created position structure.

(*type*=`float` or `None`)

Return Value

a new position

(*type*=`position_base` or a subclass)

prms(*self*)

The result of this function is to be handed to `problem_definition.logp()`. This result must contain all the information specifying the position. Normally, this is a vector of floats, but concievably, you could include other information.

vec(*self*)

Returns a numpy vector, for mathematical purposes. The result should contain all the information specifying the position; if not all, it should at least contain all the information that can be usefully expressed as a vector of floating point numbers.

Normally, `self.vec()` and `self.prms()` are identical.

__repr__(*self*)

`repr(x)`

Overrides: `object.__repr__` `exitit`(inherited documentation)

__cmp__(*self*, *other*)

This is used when the archive is sorted.

uid(*self*)

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

74.4.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

74.5 Class `position_repeatable`

This is for the common case where `logp` is a well-behaved function of its arguments. It caches positions and their corresponding values of `log(P)`.

74.5.1 Methods

<code>__init__(self, x, problem_def, logp=None)</code>
You need this function and this signature if you are going to pass it to <code>make_list_of_positions()</code> , but not necessarily otherwise.
Parameters
<code>x</code>: <i>(type=<code>numpy.ndarray</code>)</i>
<code>logp</code>: the value of <code>log(P)</code> at <code>x</code> , or <code>None</code> to indicate that it hasn't been computed yet. <i>(type=<code>float</code> or <code>None</code>)</i>
<code>problem_def</code>: <i>(type=<code>problem_definition</code> or a subclass thereof.)</i>
Raises
<code>ValueError</code> if sanity check is failed.
<code>NotGoodPosition</code> from inside <code>problem_definition.logp</code> .
Overrides: <code>object.__init__</code>

invalidate_cache(*self*)

This can be called when the mapping between parameters (x) and value changes. You might use it if you wanted to change the probability distribution (i.e. $\log(P)$).

logp(*self*)

Compute the log of the probability for the position.

Overrides: *gmisclib.mcmc.position_base.logp* extit(inherited documentation)

logp_nocompute(*self*)

Shows a recent logp value. It does not compute a value unless none has ever been computed.

Overrides: *gmisclib.mcmc.position_base.logp_nocompute* extit(inherited documentation)

new(*self*, *shift*, *logp*=None)

Returns a new position, shifted by the specified amount.

Parameters

shift: How much of a move to make to the new position.

logp: (optional) If this is supplied, is is used to set the $\log(P)$ value for the newly created position structure.

Return Value

a new position

(*type=position_base or a subclass*)

Overrides: *gmisclib.mcmc.position_base.new*

__repr__(*self*)

repr(x)

Overrides: *object.__repr__* extit(inherited documentation)

__cmp__(*self*, *other*)

This is used when the archive is sorted.

prms(*self*)

The result of this function is to be handed to `problem_definition.logp()`. This result must contain all the information specifying the position. Normally, this is a vector of floats, but conceivably, you could include other information.

uid(*self*)**vec**(*self*)

Returns a numpy vector, for mathematical purposes. The result should contain all the information specifying the position; if not all, it should at least contain all the information that can be usefully expressed as a vector of floating point numbers.

Normally, `self.vec()` and `self.prms()` are identical.

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

74.5.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

74.5.3 Class Variables

Name	Description
EPS	Value: 1e-07
HUGE	Value: 1e+38
CACHE_LIFE	Value: 50

74.6 Class `position_nonrepeatable`

This is for the (unfortunately common) case where `logp` is an independent random function of its arguments. It does not cache as much as `position_repeatable`. Arguments are analogous to `position_repeatable`.

74.6.1 Methods

`__init__(self, x, problem_def, logp=None)`

You need this function and this signature if you are going to pass it to `make_list_of_positions()`, but not necessarily otherwise.

Overrides: `object.__init__` `exitit`(inherited documentation)

`logp(self)`

Compute the log of the probability for the position.

Overrides: `gmislib.mcmc.position_base.logp` `exitit`(inherited documentation)

`logp_nocompute(self)`

Shows a recent `logp` value. It does not compute a value unless none has ever been computed.

Overrides: `gmislib.mcmc.position_base.logp_nocompute` `exitit`(inherited documentation)

`new(self, shift, logp=None)`

Returns a new position, shifted by the specified amount.

Parameters

`shift`: How much of a move to make to the new position.

`logp`: (optional) If this is supplied, it is used to set the `log(P)` value for the newly created position structure.

Return Value

a new position

(*type=***`position_base`** or a subclass)

Overrides: `gmislib.mcmc.position_base.new`

`__repr__(self)`

`repr(x)`

Overrides: `object.__repr__` `exitit`(inherited documentation)

<code>--cmp--(self, other)</code>
--

This is used when the archive is sorted.
--

<code>prms(self)</code>

The result of this function is to be handed to <code>problem_definition.logp()</code> . This result must contain all the information specifying the position. Normally, this is a vector of floats, but conceivably, you could include other information.

<code>uid(self)</code>

<code>vec(self)</code>

Returns a numpy vector, for mathematical purposes. The result should contain all the information specifying the position; if not all, it should at least contain all the information that can be usefully expressed as a vector of floating point numbers.
--

Normally, <code>self.vec()</code> and <code>self.prms()</code> are identical.

Inherited from object

`--delattr--()`, `--format--()`, `--getattr--()`, `--hash--()`, `--new--()`, `--reduce--()`, `--reduce_ex--()`, `--setattr--()`, `--sizeof--()`, `--str--()`, `--subclasshook--()`

74.6.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

74.6.3 Class Variables

Name	Description
HUGE	Value: 1e+38

74.7 Class `position_history_dependent`



This is for the case where `logp` is a history-dependent function of its arguments. This is the most general, most expensive case.

74.7.1 Methods

`__init__(self, x, problem_def, logp=None)`

You need this function and this signature if you are going to pass it to `make_list_of_positions()`, but not necessarily otherwise.

Overrides: `object.__init__` extit(inherited documentation)

`logp(self)`

Compute the log of the probability for the position.

Overrides: `gmisclib.mcmc.position_base.logp` extit(inherited documentation)

`logp_nocompute(self)`

Shows a recent `logp` value. It does not compute a value unless none has ever been computed.

Overrides: `gmisclib.mcmc.position_base.logp_nocompute` extit(inherited documentation)

new(*self*, *shift*, *logp*=None)

Returns a new position, shifted by the specified amount.

Parameters

shift: How much of a move to make to the new position.

logp: (optional) If this is supplied, is is used to set the log(P) value for the newly created position structure.

Return Value

a new position

(*type*=**position_base** or a subclass)

Overrides: *gmisclib.mcmc.position_base.new*

__repr__(*self*)

repr(x)

Overrides: object.__repr__ exitit(inherited documentation)

__cmp__(*self*, *other*)

This is used when the archive is sorted.

prms(*self*)

The result of this function is to be handed to *problem_definition.logp()*. This result must contain all the information specifying the position. Normally, this is a vector of floats, but concievably, you could include other information.

uid(*self*)

vec(*self*)

Returns a numpy vector, for mathematical purposes. The result should contain all the information specifying the position; if not all, it should at least contain all the information that can be usefully expressed as a vector of floating point numbers.

Normally, *self.vec()* and *self.prms()* are identical.

Inherited from object

__delattr__(), *__format__()*, *__getattr__()*, *__hash__()*, *__new__()*, *__reduce__()*, *__reduce_ex__()*, *__setattr__()*, *__sizeof__()*, *__str__()*, *__subclasshook__()*

74.7.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

74.7.3 Class Variables

Name	Description
<code>HUGE</code>	Value: 1e+38

74.8 Class `T_acceptor`



This class implements a normal Metropolis-Hastings acceptance of steps.

74.8.1 Methods

<code>__init__</code> (<i>self</i> , <i>T</i> =1.0)
You can change the temperature to do simulated annealing.
Parameters
<i>T</i> : temperature
(<i>type</i> = <i>float</i>)
Overrides: <code>object.__init__</code>

<code>T</code> (<i>self</i>)
Return Value
the system temperature.
(<i>type</i> = <i>float</i>)

```
--call--(self, delta)
```

Accept a step or not?

Parameters

delta: The proposed step gives **delta** as the change in **log(probability)**.

(*type=float*)

Return Value

should it be accepted or not?

(*type=bool*)

Inherited from object

```
--delattr--(), --format--(), --getattr__(), --hash--(), --new--(), --reduce--(), --reduce_ex--(),
--repr--(), --setattr--(), --sizeof--(), --str--(), --subclasshook--()
```

74.8.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

74.9 Class stepper

```
object └─
          gmisclib.mcmc.stepper
```

Known Subclasses: gmisclib.mcmc.BootStepper

This is your basic stepper class that incrementally will give you a Markov-Chain Monte-Carlo series of samples from a probability distribution.

74.9.1 Methods

```
--init--(self)
```

`x.__init__(...)` initializes x; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` extit(inherited documentation)

step(*self*)

In subclasses, this takes a step and returns 0 or 1, depending on whether the step was accepted or not.

prms(*self*)

Return Value

The current parameters

(*type=*`numpy.ndarray`)

prmlist(*self*)

Returns all stored positions.

status(*self*)

Provides some printable status information in a=v; format.

reset(*self*)

Called internally to mark when the optimization has found a new minimum. [NOTE: You might also call it if the function you are minimizing changes.]

reset_id(*self*)

Use this to tell if the stepper has been reset since you last looked at it.

needs_a_reset(*self*)

Decides if we we need a reset. This checks to see if we have a new record logP that exceeds the old record. It keeps track of the necessary paperwork.

current(*self*)

Return Value

the current position.

(*type=*`position_base`)

Inherited from object

`--delattr__()`, `--format__()`, `--getattr__()`, `--hash__()`, `--new__()`, `--reduce__()`, `--reduce_ex__()`, `--repr__()`, `--setattr__()`, `--sizeof__()`, `--str__()`, `--subclasshook__()`

74.9.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

74.10 Class adjuster

object —
 gmisclib.mcmc.adjuster

74.10.1 Methods

__init__(*self*, *F*, *vscale*, *vse*=0.0, *vsmax*=1e+30)

x.**__init__**(...) initializes *x*; see *x*.**__class__**.**__doc__** for signature

Overrides: object.**__init__** **exitit**(inherited documentation)

reset(*self*)

f(*self*)

This allows the desired fraction of accepted steps to depend on *self.vscale*.

Note: This method should only be called with **self.vse** != 0 where *self.vscale* can normally be expected to be fairly close to unity, and where small values of **self.vse** indicate trouble. In the **mcmc.Bootstepper.step_boot** case this is true.

inctry(*self*, *accepted*)

vs(*self*)

We stick in the factor of **random.lognormvariate()** so that all sizes of move are possible and thus we can prove that we can random-walk to any point in a connected region. This makes the proof of ergodicity simpler.

status(*self*)

Inherited from object

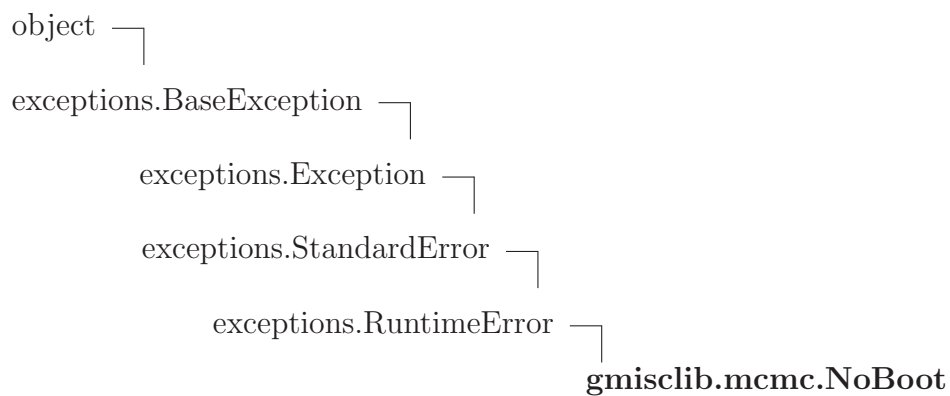
__delattr__(), **__format__**(), **__getattr__**(), **__hash__**(), **__new__**(), **__reduce__**(), **__reduce_ex__**(), **__repr__**(), **__setattr__**(), **__sizeof__**(), **__str__**(), **__subclasshook__**()

74.10.2 Properties

Name	Description
<i>Inherited from object</i> __class__	

74.10.3 Class Variables

Name	Description
Z0	Value: 2.30258509299
DZ	Value: 0.3
TOL	Value: 0.2
STABEXP	Value: 1.0

74.11 Class NoBoot**74.11.1 Methods**

__init__(self) x.__init__(...) initializes x; see x.__class__.__doc__ for signature Overrides: object.__init__ extit(inherited documentation)
--

Inherited from exceptions.RuntimeError

`--new--()`

Inherited from exceptions.BaseException

`--delattr--()`, `--getattrattribute--()`, `--getitem--()`, `--getslice--()`, `--reduce--()`, `--repr--()`,

`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

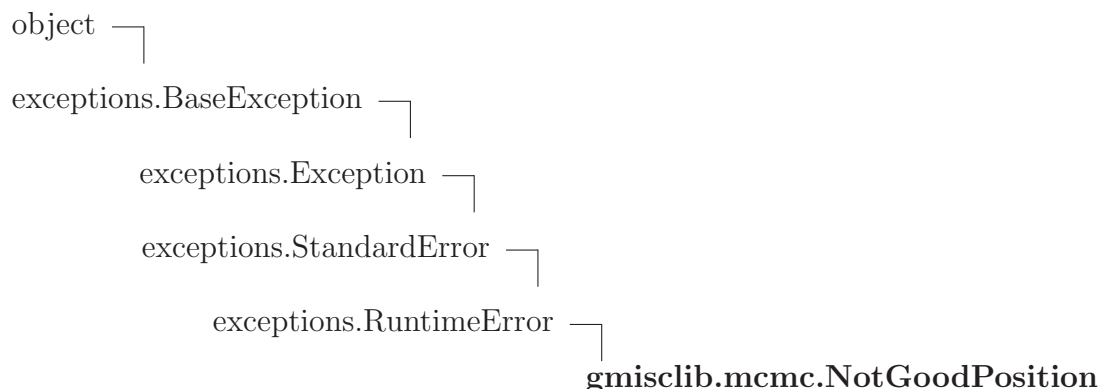
Inherited from object

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

74.11.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
<code>args</code> , <code>message</code>	
<i>Inherited from object</i>	
<code>__class__</code>	

74.12 Class `NotGoodPosition`



74.12.1 Methods

<code>__init__(self, *s)</code>
<code>x.__init__(...)</code> initializes <code>x</code> ; see <code>x.__class__.__doc__</code> for signature
Overrides: <code>object.__init__</code> extit(inherited documentation)

Inherited from `exceptions.RuntimeError`

`__new__()`

Inherited from `exceptions.BaseException`

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

Inherited from object

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

74.12.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
<code>args</code> , <code>message</code>	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

74.13 Class `hashcounter_c`**74.13.1 Methods**

<code>incr(self, x)</code>

<code>decr(self, x)</code>

Inherited from dict

`__cmp__()`, `__contains__()`, `__delitem__()`, `__eq__()`, `__ge__()`, `__getattr__()`, `__getitem__()`, `__gt__()`, `__init__()`, `__iter__()`, `__le__()`, `__len__()`, `__lt__()`, `__ne__()`, `__new__()`, `__repr__()`, `__setitem__()`, `__sizeof__()`, `clear()`, `copy()`, `fromkeys()`, `get()`, `has_key()`, `items()`, `iteritems()`, `iterkeys()`, `itervalues()`, `keys()`, `pop()`, `popitem()`, `setdefault()`, `update()`, `values()`

Inherited from object

`__delattr__()`, `__format__()`, `__reduce__()`, `__reduce_ex__()`, `__setattr__()`, `__str__()`, `__subclasshook__()`

74.13.2 Properties

Name	Description
<i>Inherited from <code>object</code></i>	

continued on next page

Name	Description
<code>__class__</code>	

74.13.3 Class Variables

Name	Description
<i>Inherited from dict</i>	
<code>__hash__</code>	

74.14 Class Archive

object —
 gmisclib.mcmc.Archive

Known Subclasses: gmisclib.mcmc.ContPrmArchive

This maintains a list of all the recent accepted positions.

74.14.1 Methods

`__init__(self, lop, np_eff, sort='auto', maxdups=-1, maxArchSize=None, alpha=None)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` `exitit` (inherited documentation)

`distinct_count(self)`

How many distinct values of the parameters are there in the archive?

`prmlist(self)`

`sort(self)`

Called under lock. Sort the archive into order of `logp`.

`reset(self)`

We sort the archive to speed the convergence to the best solution. After all, if you've just gotten a reset, it is likely that you're not at the bottom yet, so statistical properties of the distribution are likely to be irrelevant.

<code>--len--(self)</code>

<code>choose(self)</code>

<code>truncate(self, desired_length)</code>

Desired length is in terms of the number of distinct positions. Called under lock.
--

<code>append(self, x, forbid_dups)</code>

Adds stuff to the archive, possibly sorting the new information into place. It updates all kinds of counters and summary statistics.
--

Parameters

<code>x:</code>	A position to (possibly) add to the archive. (<i>type=position.base</i>)
-----------------	---

<code>forbid_dups:</code>	Don't add duplicates to the archive. (<i>type=boolean</i>)
---------------------------	---

Return Value

A one-letter code indicating what happened. 'f' if x is a duplicate and duplicates are forbidden. 'd' if it is a duplicate and there have been too many duplicates lately. 'a' otherwise – x has been added to the archive. (<i>type=str</i>)
--

<code>append_hook(self, x)</code>

<code>truncate_hook(self, to_be_dropped)</code>

Inherited from object

`--delattr--()`, `--format--()`, `--getattribute--()`, `--hash--()`, `--new--()`, `--reduce--()`, `--reduce_ex--()`, `--repr--()`, `--setattr--()`, `--sizeof--()`, `--str--()`, `--subclasshook--()`

74.14.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

74.14.3 Class Variables

Name	Description
SSALWAYS	Value: 'always'
SSNEVER	Value: 'neversort'
SSAUTO	Value: 'auto'
SSLOW	Value: 'startlow'

74.15 Class *ContPrmArchive*

object └

gmisclib.mcmc.Archive └

gmisclib.mcmc.ContPrmArchive

74.15.1 Methods

`__init__(self, lop, np_eff, sort='auto', maxdups=-1, maxArchSize=None, alpha=None)`

Append_hook() is called for every element of the archive. That function can be replaced in a sub-class to accumulate some kind of summary. Here, it is used to keep track of parameter means and standard deviations.

Overrides: object.__init__

`append_hook(self, x)`

This accumulates parameter means and standard deviations.

Overrides: gmisclib.mcmc.Archive.append_hook

`truncate_hook(self, to_be_dropped)`

Overrides: gmisclib.mcmc.Archive.truncate_hook

`variance(self)`

`__len__(self)`

append(*self*, *x*, *forbid_dups*)

Adds stuff to the archive, possibly sorting the new information into place. It updates all kinds of counters and summary statistics.

Parameters

x: A position to (possibly) add to the archive.
(type=position.base)

forbid_dups: Don't add duplicates to the archive.
(type=boolean)

Return Value

A one-letter code indicating what happened. 'f' if x is a duplicate and duplicates are forbidden. 'd' if it is a duplicate and there have been too many duplicates lately. 'a' otherwise – x has been added to the archive.

(type=str)

choose(*self*)

distinct_count(*self*)

How many distinct values of the parameters are there in the archive?

prmlist(*self*)

reset(*self*)

We sort the archive to speed the convergence to the best solution. After all, if you've just gotten a reset, it is likely that you're not at the bottom yet, so statistical properties of the distribution are likely to be irrelevant.

sort(*self*)

Called under lock. Sort the archive into order of `logp`.

truncate(*self*, *desired_length*)

Desired length is in terms of the number of distinct positions. Called under lock.

Inherited from object

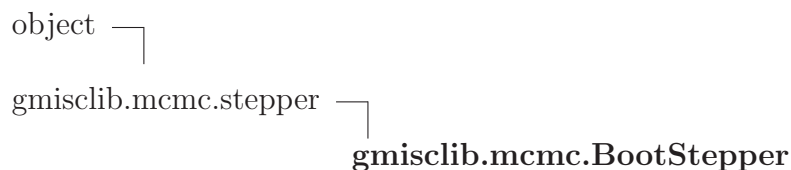
`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

74.15.2 Properties

Name	Description
<i>Inherited from object</i> __class__	

74.15.3 Class Variables

Name	Description
SSALWAYS	Value: 'always'
SSAUTO	Value: 'auto'
SSLOW	Value: 'startlow'
SSNEVER	Value: 'neversort'

74.16 Class BootStepper

Known Subclasses: *gmislib.mcmc.big.BootStepper*

74.16.1 Methods

__init__(self, lop, v, sort='auto', maxArchSize=None, parallelSizeDiv=1)

x.__init__(...) initializes *x*; see *x.__class__.__doc__* for signature

Parameters

maxArchSize: How many position vectors can be stored. This is normally used to (loosely) enforce a memory limitation for large jobs.

parallelSizeDiv: For use when there are several cooperating MCMC processes that share data. When >1, this allows each process to have smaller stored lists. Normally, parallelSizeDiv is between 1 and the number of cooperating processes.

Overrides: *object.__init__*

step(*self*)

In subclasses, this takes a step and returns 0 or 1, depending on whether the step was accepted or not.

Overrides: *gmisclib.mcmc.stepper.step* *exitit*(inherited documentation)

prmlist(*self*)

Returns all stored positions.

Overrides: *gmisclib.mcmc.stepper.prmlist* *exitit*(inherited documentation)

status(*self*)

Provides some printable status information in a=v; format.

Overrides: *gmisclib.mcmc.stepper.status* *exitit*(inherited documentation)

stepV(*self*)**step_boot**(*self*)**reset_adjusters**(*self*)**ergodic**(*self*)

A crude measure of how ergodic the MCMC is.

Return Value

The inverse of how many steps it takes to cross the minimum in the slowest direction. Or zero, if it is too soon after some major violation of the MCMC assumptions.

set_sort_strategy(*self*, *ss*)**current**(*self*)**Return Value**

the current position.

(*type=position_base*)

needs_a_reset(*self*)

Decides if we we need a reset. This checks to see if we have a new record logP that exceeds the old record. It keeps track of the necessary paperwork.

prms(*self*)

Return Value

The current parameters

(*type=*`numpy.ndarray`)

reset(*self*)

Called internally to mark when the optimization has found a new minimum.
[NOTE: You might also call it if the function you are minimizing changes.]

reset_id(*self*)

Use this to tell if the stepper has been reset since you last looked at it.

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

74.16.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

74.16.3 Class Variables

Name	Description
F	Value: 0.234
alpha	Value: 0.1
PBootLim	Value: 0.9
SSAUTO	Value: 'auto'
SSNEVER	Value: 'neversort'
SSLOW	Value: 'startlow'
SSALWAYS	Value: 'always'

75 Module **gmisclib.mcmcS2**

Markov-Chain Monte-Carlo algorithms.

Here, we do MCMC when $-\log(p)$ is a sum of squares.

75.1 Functions

<code>go_step(<i>x</i>, <i>showvec</i>)</code>
--

76 Module *gmislib.mcmcSQ*

Markov-Chain Monte-Carlo algorithms.

Here, we do MCMC when $-\log(p)$ is a sum of squares.

Version: \$Revision: 1.62 \$

76.1 Functions

linear_least_squares_lambda (<i>a</i> , <i>b</i> , <i>lambda</i>)
--

test_llsl_1 ()

Simplest case - one parameter, one observation.

test_llsl_1o ()

Overdetermined.

test_llsl_2u ()

Underdetermined.

test_llsl_2 ()

Simplest case - two parameters, two observations.

test_stepper ()

get_localfit (<i>selectprms</i>)

test_get_localfit ()

sigma (<i>x</i>)

get_1_localfit (<i>selectprms</i>)

test_llsl ()

test ()

diag_variance (<i>start</i>)

go(*argv*)

Run an optimization from command line flags.

76.2 Variables

Name	Description
VARFUDGE	Value: 0.707
VF2	Value: 1.2
EFUDGE	Value: 1.0/ 2.0
WORRY	Value: 6
MINIMIZE	Value: 0.0
D	Value: 1
SINGPASSES	Value: 10

76.3 Class *CantGetPoint*

object └

exceptions.BaseException └

exceptions.Exception └

exceptions.StandardError └

exceptions.RuntimeError └

gmislib.mcmcSQ.CantGetPoint

76.3.1 Methods

__init__(*self*, *s*)

x.**__init__**(...) initializes *x*; see *x*.**__class__**.**__doc__** for signature

Overrides: object.**__init__** extit(inherited documentation)

Inherited from exceptions.RuntimeError

__new__()

Inherited from exceptions.BaseException

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

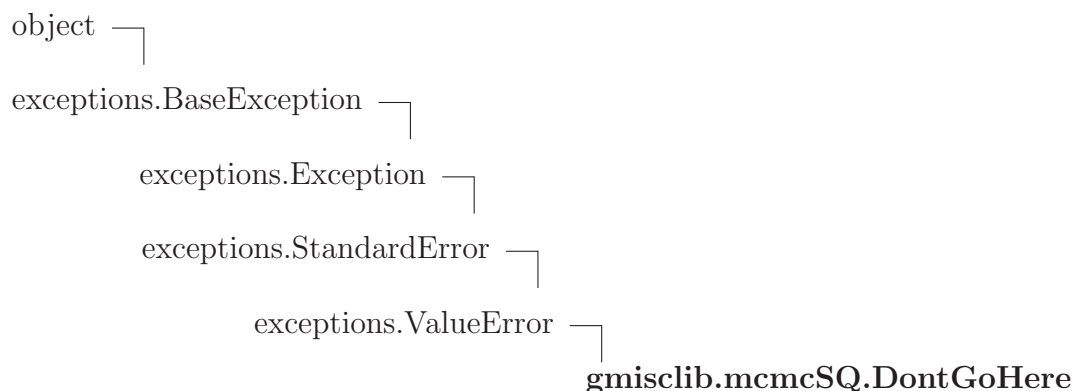
Inherited from object

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

76.3.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
args, message	
<i>Inherited from object</i>	
<code>__class__</code>	

76.4 Class DontGoHere



76.4.1 Methods

`__init__(self, s)`
 x.`__init__`(...) initializes x; see x.`__class__`.`__doc__` for signature
 Overrides: object.`__init__` extit(inherited documentation)

Inherited from exceptions.ValueError

`__new__()`

Inherited from exceptions.BaseException

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,

`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

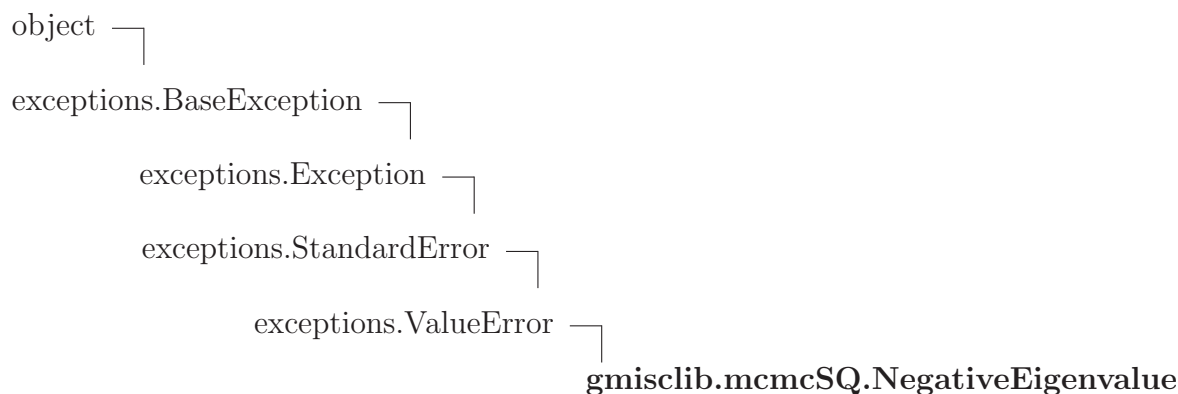
Inherited from object

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

76.4.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
args, message	
<i>Inherited from object</i>	
__class__	

76.5 Class NegativeEigenvalue



76.5.1 Methods

<code>__init__(self, ev)</code>
x. <code>__init__</code> (...) initializes x; see x. <code>__class__</code> . <code>__doc__</code> for signature
Overrides: object. <code>__init__</code> extit(inherited documentation)

Inherited from exceptions.ValueError

`__new__()`

Inherited from exceptions.BaseException

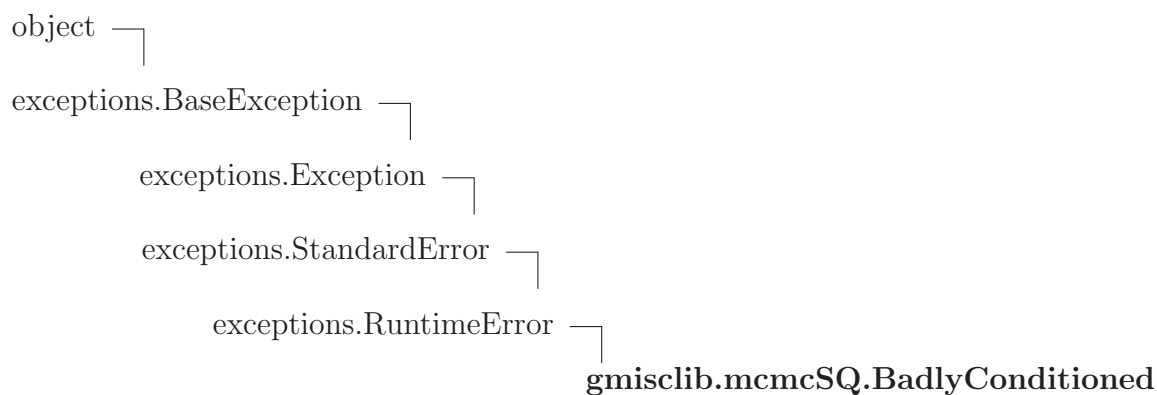
`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

Inherited from object

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

76.5.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
<code>args</code> , <code>message</code>	
<i>Inherited from object</i>	
<code>__class__</code>	

76.6 Class *BadlyConditioned***76.6.1 Methods**

`__init__(self, s)`
`x.__init__(...)` initializes x; see `x.__class__.__doc__` for signature
 Overrides: `object.__init__` extit(inherited documentation)

Inherited from exceptions.RuntimeError

`__new__()`

Inherited from exceptions.BaseException

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

Inherited from object

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

76.6.2 Properties

Name	Description
<code>__init__</code>	Inherited from <code>exceptions.BaseException</code> args, message
<code>__class__</code>	Inherited from object

76.7 Class timer

76.7.1 Methods

<code>__init__(self)</code>
<code>start(self)</code>
<code>stop(self)</code>
<code>is_running(self)</code>
<code>running_total(self)</code>
<code>get(self)</code>
<code>leak(self, fac=0.8)</code>

76.8 Class problem_definition

object —
 gmisclib.mcmcSQ.problem_definition

76.8.1 Methods

```
__init__(self, resid_fcn, c, fixer=None)
```

x.**__init__**(...) initializes *x*; see *x*.**__class__**.**__doc__** for signature

Overrides: *object*.**__init__** extit(inherited documentation)

```
resid(self, x)
```

```
fixer(self, x)
```

This allows for symmetries in problems.

Inherited from object

__delattr__(), **__format__**(), **__getattr__**(), **__hash__**(), **__new__**(), **__reduce__**(), **__reduce_ex__**(), **__repr__**(), **__setattr__**(), **__sizeof__**(), **__str__**(), **__subclasshook__**()

76.8.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

76.9 Class position

```
object └─ gmisclib.mcmcSQ.position
```

76.9.1 Methods

```
__init__(self, x, problem_def, resid=None)
```

x.**__init__**(...) initializes *x*; see *x*.**__class__**.**__doc__** for signature

Overrides: *object*.**__init__** extit(inherited documentation)

```
attach_timer(self, tmr)
```

```
resid(self)
```

<code>badspot(self)</code>

<code>sumsq(self)</code>

<code>logp(self)</code>

<code>shape(self)</code>

<code>new(self, shift)</code>

Returns a new position, shifted by the specified amount.
--

<code>prms(self)</code>

<code>--str--(self)</code>

<code>str(x)</code>

Overrides: <code>object.__str__</code> extit(inherited documentation)

Inherited from object

`--delattr--()`, `--format--()`, `--getattr--()`, `--hash--()`, `--new--()`, `--reduce--()`, `--reduce_ex--()`, `--repr--()`, `--setattr--()`, `--sizeof--()`, `--subclasshook--()`

76.9.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

76.9.3 Class Variables

Name	Description
HUGE	Value: 1e30

76.10 Class *Lambda_c*

76.10.1 Methods

`__init__(self, lmbda, np)`

lmbda is the initial value; *np* is the number of parameters, which is used to control whether adjustments are fine (late) or coarse (early).

`adjust_BadEstKluge(self, f)`

`adjust_VfacKluge(self, vfac)`

`fa(self)`

`fr(self)`

`lmbda(self)`

Return the current value of *lmbda*.

`neg(self, min_eigenval)`

`sing(self)`

`accept(self)`

`reject(self)`

`low_lim(self, x)`

`high_lim(self, x)`

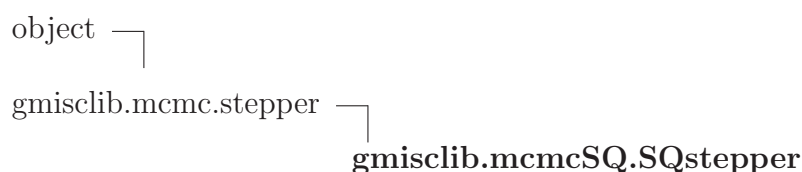
76.10.2 Class Variables

Name	Description
<code>__doc__</code>	Value: ""This class remembers and increments Lamb...
<code>EARLY_REJECT</code>	Value: 10.0
<code>EARLY_ACCEPT</code>	Value: 0.1
<code>ACCEPT</code>	Value: 0.2

continued on next page

Name	Description
LATE_ACCEPT	Value: 0.6
AR_RATIO	Value: 0.25
MIN	Value: 1e-10
NEG	Value: 5.0
SING	Value: 2.0
LIM_F	Value: 10.0

76.11 Class SQstepper



76.11.1 Methods

__init__(*self*, *pd*, *lop*, *v*, *minimize*=0)

x.__init__(...) initializes *x*; see *x*.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

sumsq(*self*)

current(*self*)

Return Value

the current position.

(*type*=*position_base*)

Overrides: gmisclib.mcmc.stepper.current extit(inherited documentation)

last(*self*)

trypos(*self*, *newx*, *here*)

step(*self*, *evmisc*=None, *atmisc*=None, *showvec*=0)

Returns 0 or 1, depending on whether the step was accepted or not.

Overrides: gmisclib.mcmc.stepper.step

status(*self*)

Provides some printable status information in a=v; format.

Overrides: gmisclib.mcmc.stepper.status extit(inherited documentation)

print_ev_diag(*self*, *misc*=None, *vectors*=0)

print_at(*self*, *misc*=None)

Prints a file to match the old C++ optimizer output.

print_rejected(*self*, *r*, *misc*=None)

needs_a_reset(*self*)

Decides if we we need a reset. This checks to see if we have a new record logP that exceeds the old record. It keeps track of the necessary paperwork.

prmlist(*self*)

Returns all stored positions.

prms(*self*)

Return Value

The current parameters

(*type*=*numpy.ndarray*)

reset(*self*)

Called internally to mark when the optimization has found a new minimum.
[NOTE: You might also call it if the function you are minimizing changes.]

reset_id(*self*)

Use this to tell if the stepper has been reset since you last looked at it.

Inherited from object

`--delattr__()`, `--format__()`, `--getattr__()`, `--hash__()`, `--new__()`, `--reduce__()`, `--reduce_ex__()`,
`--repr__()`, `--setattr__()`, `--sizeof__()`, `--str__()`, `--subclasshook__()`

76.11.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

76.12 Class archiver

76.12.1 Methods

<code>--init--(self, np, v, pd, fcntime)</code>

<code>trim_initial(self, steps)</code>
--

<code>get_new(self)</code>

<code>widen(self)</code>

<code>narrow(self)</code>

<code>choose_guts(self)</code>

<code>choose(self)</code>

<code>set_current(self, setting)</code>

<code>current(self)</code>

<code>last(self)</code>

<code>go(self)</code>

<code>add_initial(self, x)</code>

Returns 1 once there is enough data. Otherwise returns 0.

add(*self*, *x*, *location=-1*)

Add a point to the history list if it is generally good. Location tells you whether to add to the beginning (0) or end (-1) of the history list. If adding on the end, if the new point is very close to a previous point on the list, delete the previous point. If adding to the beginning, only add if there is no close point later in the list. It returns the net number of points added.

describe(*self*)

76.12.2 Class Variables

Name	Description
<code>__doc__</code>	Value: ...
<code>SMALL_VOL</code>	Value: 0.1
<code>SMALL_PRE</code>	Value: 0.5

77 Module *gmisclib.mcmc_big*

An extension of *mcmc* that includes a new stepping algorithm.

77.1 Functions

pairmax (* <i>xy</i>)

Finds the (x,y) pair which has the largest y.

test ()

find_closest_p (<i>v</i> , * <i>vp</i>)
--

Searches a list of (v,p) pairs and finds the one whose v is closest to the first argument. Returns (v,p) of the closest pair.

bootstepper (<i>logp</i> , <i>x</i> , <i>v</i> , <i>c=None</i> , <i>sort='auto'</i> , <i>fixer=None</i> , <i>repeatable=True</i>)
--

This is (essentially) another interface to the class constructor. It's really there for backwards compatibility.
--

77.2 Variables

Name	Description
<code>--package--</code>	Value: 'gmisclib'

77.3 Class *BootStepper*

object └

gmisclib.mcmc.stepper └

gmisclib.mcmc.BootStepper └
 gmisclib.mcmc_big.BootStepper

77.3.1 Methods

`__init__(self, lop, v, sort='auto', maxArchSize=None, parallelSizeDiv=1)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Parameters

`maxArchSize`: How many position vectors can be stored. This is normally used to (loosely) enforce a memory limitation for large jobs.

`parallelSizeDiv`: For use when there are several cooperating MCMC processes that share data. When >1 , this allows each process to have smaller stored lists. Normally, `parallelSizeDiv` is between 1 and the number of cooperating processes.

Overrides: `object.__init__` `exitit`(inherited documentation)

`step(self)`

In subclasses, this takes a step and returns 0 or 1, depending on whether the step was accepted or not.

Overrides: `gmisclib.mcmc.stepper.step` `exitit`(inherited documentation)

`reset_adjusters(self)`

Overrides: `gmisclib.mcmc.BootStepper.reset_adjusters`

`step_mixed(self)`

`step_parab(self)`

`current(self)`

Return Value

the current position.

(*type=position_base*)

ergodic(*self*)

A crude measure of how ergodic the MCMC is.

Return Value

The inverse of how many steps it takes to cross the minimum in the slowest direction. Or zero, if it is too soon after some major violation of the MCMC assumptions.

needs_a_reset(*self*)

Decides if we we need a reset. This checks to see if we have a new record logP that exceeds the old record. It keeps track of the necessary paperwork.

prmlist(*self*)

Returns all stored positions.

Overrides: *gmisclib.mcmc.stepper.prmlist* *exitit*(inherited documentation)

prms(*self*)**Return Value**

The current parameters
(*type=**numpy.ndarray*)

reset(*self*)

Called internally to mark when the optimization has found a new minimum. [NOTE: You might also call it if the function you are minimizing changes.]

reset_id(*self*)

Use this to tell if the stepper has been reset since you last looked at it.

set_sort_strategy(*self*, *ss*)**status(*self*)**

Provides some printable status information in a=v; format.

Overrides: *gmisclib.mcmc.stepper.status* *exitit*(inherited documentation)

stepV(*self*)**step_boot(*self*)**

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

77.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

77.3.3 Class Variables

Name	Description
F	Value: 0.234
PBootLim	Value: 0.9
SSALWAYS	Value: 'always'
SSAUTO	Value: 'auto'
SSLOW	Value: 'startlow'
SSNEVER	Value: 'neversort'
alpha	Value: 0.1

78 Module `gmisclib.mcmc_cooperate`

78.1 Functions

<code>test0()</code>

<code>test0s()</code>

<code>test1()</code>

<code>test_many(<i>n</i>, <i>fcn</i>)</code>
--

<code>op_string_median(<i>tmp</i>)</code>

<code>op_float_median(<i>tmp</i>)</code>
--

78.2 Variables

Name	Description
encoder	Value: <code>g_encode.encoder(regex=r"^[a-zA-Z0-9<>?,./:~;'\{\}[\]\!@\$...)</code>
Ops	Value: <code>{'string_median': op_string_median, 'float_median': op_fl...</code>
Unpack	Value: <code>{'float_median': <type 'float'>, 'string_median': <type '...</code>
test_args	Value: <code>('localhost', 8487, 'K', 'job')</code>
<code>--package--</code>	Value: <code>'gmisclib'</code>

78.3 Class Barrier

object — `gmisclib.mcmc_cooperate.Barrier`

78.3.1 Methods**`__init__(self, *x)`**

Constructs a barrier from a list of integers.

Overrides: `object.__init__`**`__cmp__(self, other)`****`__iadd__(self, other)`****`__repr__(self)`**`repr(x)`Overrides: `object.__repr__` `exitit`(inherited documentation)**`deepen(self, v)`*****Inherited from object***`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`**78.3.2 Properties**

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

78.4 Class Oops

object └

exceptions.BaseException └

exceptions.Exception └

`gmislib.mcmc_cooperate.Oops`**Known Subclasses:** `gmislib.mcmc_cooperate.LateToBarrier`

78.4.1 Methods

`__init__`(*self*, **s*)

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` extit(inherited documentation)

Inherited from exceptions.Exception`__new__()`*Inherited from exceptions.BaseException*

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

Inherited from object

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

78.4.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
<code>args</code> , <code>message</code>	
<i>Inherited from object</i>	
<code>__class__</code>	

78.5 Class *LateToBarrier*

78.5.1 Methods

`__init__(self, *s)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` `exitit`(inherited documentation)

Inherited from `exceptions.Exception`

`__new__()`

Inherited from `exceptions.BaseException`

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

Inherited from `object`

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

78.5.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
<code>args</code> , <code>message</code>	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

78.6 Class connection

object —
 gmisclib.mcmc_cooperate.connection

78.6.1 Methods

`__init__(self, host, port, Key, jobid)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` `exitit`(inherited documentation)

`send(self, *s)`

```
flush(self)
```

```
recv(self)
```

```
list_ops(self)
```

```
version(self)
```

```
rank(self)
```

Return Value

number_of_processes, rank_of_this_process
(*type*=(int, int))

```
close(self)
```

```
__del__(self)
```

```
barrier(self, b, nmin=0, exc=True)
```

```
swap_vec(self, logp, v)
```

Return Value

(logp, vector)

```
set(self, *kv)
```

```
get_list(self, key)
```

```
get_combined(self, key, operation)
```

```
spread(self, key, value, barrier, nmin=0)
```

```
get_consensus(self, key, value, barrier, operation, nmin=0)
```

Inherited from object

```
__delattr__(), __format__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),  
__repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()
```

78.6.2 Properties

Name	Description
<i>Inherited from object</i> __class__	

78.6.3 Class Variables

Name	Description
e	Value: g_encode.encoder(regex= r""""[[^] a-zA-Z0-9<>?,./: ";'{}[\]!@\$...

79 Module *gmisclib.mcmc_helper*

This is a helper module to make use of *mcmc.py* and *mcmc_big.py*. It allows you to conveniently run a Monte-Carlo simulation of any kind until it converges (*stepper.run_to_bottom*) or until it has explored a large chunk of parameter space (*stepper.run_to_ergodic*).

It also helps you with logging the process.

79.1 Functions

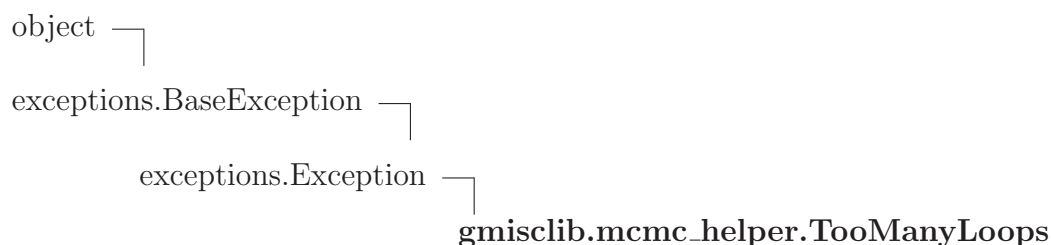
```
make_stepper_from_lov(problem_def, list_of_vectors, mcmc_mod=<module
'gmisclib.mcmc' from '/home/gpk/n/speechresearch/...',
posn_class=<class 'gmisclib.mcmc.position_repeatable'>)
```

```
test()
```

79.2 Variables

Name	Description
Debug	Value: 0
__package__	Value: 'gmisclib'

79.3 Class *TooManyLoops*



79.3.1 Methods

```
__init__(self, *s)
x.__init__(...) initializes x; see x.__class__.__doc__ for signature
Overrides: object.__init__ exitit(inherited documentation)
```

Inherited from `exceptions.Exception``__new__()`***Inherited from `exceptions.BaseException`***`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`***Inherited from `object`***`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`**79.3.2 Properties**

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
<code>args</code> , <code>message</code>	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

79.4 Class `warnevery`**79.4.1 Methods**`__init__(self, interval)``x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signatureOverrides: `object.__init__` `extit`(inherited documentation)`inc(self, incr=1)``count(self)`***Inherited from `object`***`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

79.4.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

79.5 Class logger_template

object └─ gmisclib.mcmc_helper.logger_template

Known Subclasses: gmisclib.mcmc_logger.logger_c

79.5.1 Methods

add (<i>self</i> , <i>stepperInstance</i> , <i>iter</i>)

close (<i>self</i>)

reset (<i>self</i>)

Inherited from object

__delattr__(), __format__(), __getattr__(), __hash__(), __init__(), __new__(), __reduce__(),
__reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

79.5.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

79.6 Class stepper

object └─ gmisclib.mcmc_helper.stepper

Known Subclasses: gmisclib.mcmc_socket.stepper

79.6.1 Methods

`__init__(self, x, maxloops=-1, logger=None, share=None)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` `exitit` (inherited documentation)

`reset_loops(self, maxloops=-1)`

`run_to_change(self, ncw=1, acceptable_step=None, update_T=False)`

Run the *Markov Chain Monte-Carlo* until it finds an acceptable step.

Parameters

- ncw:** How many steps should be accepted before it stops?
(*type=int*)
- acceptable_step:** A function that decides what steps are acceptable and what are not. This is passed to the MCMC stepper, `mcmc.BootStepper`.
(*type=function(float) -> bool, often `mcmc.T_acceptor` or `step_acceptor`*)
- update_T:** Obsolete. Must be False.
(*type=boolean.*)

Raises

- TooManyLoops** if it takes a long time before enough steps are accepted.

`communicate_hook(self, id)`

`close(self)`

After calling `close()`, it is no longer legal to call `run_to_change()`, `run_to_ergodic()`, or `run_to_bottom()`. Logging is shut down, but the contents of the stepper are still available for inspection.

run_to_ergodic(*self*, *ncw*=1, *T*=1.0)

Run the stepper until it has explored all of parameter space **ncw** times (as best as we can estimate). Note that this is a pretty careful routine. If the stepper finds a better region of parameter space so that log(P) improves part way through the process, the routine will reset itself and begin again.

NOTE: this sets **T** and **sortstrategy** for the **stepper**.

Parameters

ncw: how many times (or what fraction) of an ergodic exploration to make.

(*type=int.*)

T: temperature at which to run the Monte-Carlo process. This is normally 1.0. If it is **None**, then the current temperature is not modified.

(*type=float or None.*)

Return Value

a **position** at the end of the process.

run_to_bottom(*self*, *ns*=3, *acceptable_step*=None)

Run the *Markov Chain Monte-Carlo* until it converges near a minimum.

The general idea of the termination condition is that it keeps track of the angle between successive sequences of steps, where each sequence contains **ns** steps. If the angle is less than 90 degrees, it is evidence that the solution is still drifting in some consistent direction and therefore not yet at the maximum. Angles of greater than 90 degrees suggest that the optimization is wandering aimlessly near a minimum. It will run until a sufficient number of large angles are seen since the last BMCMC reset. A similar check is made on individual components of the parameter vector.

Parameters

ns: This controls how carefully it checks that it has really found a minimum. Large values will take longer but will assure more accurate convergence.

(*type*=**int**)

acceptable_step: A callable object (i.e. class or function) that returns **True** or **False**, depending on whether a proposed step is acceptable. See `mcmc.T_acceptor` for an example.

Raises

TooManyLoops ?

synchronize_start(*self*, *id*)

synchronize_end(*self*, *id*)

synchronize_abort(*self*, *id*)

note(*self*, *s*, *lvl*)

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

79.6.2 Properties

Name	Description
<i>Inherited from object</i>	

continued on next page

Name	Description
<code>--class--</code>	

79.7 Class `step_acceptor`

object —
 `gmisclib.mcmc_helper.step_acceptor`

This class defines the annealing schedule when `mcmc` is used as an optimizer. It corresponds to `n` extra degrees of freedom that exchange probability (i.e. ‘energy’) with each other, and the system to be optimized. Over time, the excess $\log(\text{probability})$ gradually oozes away.

79.7.1 Methods

`__init__(self, n, T0, ooze, T=1.0, maxT=None)`

Create a *step_acceptor* and define it's properties.

Parameters

- n:** How many degrees of freedom should the step acceptor store?
(*type=int*)
- T:** What's the temperature of the heat bath? This is the final temperature that the annealing schedule will eventually approach. Default: 1.0.
(*type=float*)
- ooze:** When a step is accepted, one of the degrees of freedom is oozed toward T. This specifies how far. After $n/(1-ooze)$ steps, these degrees of freedom will have lost most of their excess probability and the temperature will be approaching T.
(*type=float*)
- T0:** The starting temperature.
(*type=float*)
- maxT:** In optimization mode, it's normal for the system being optimized to increase it's $\log(P)$. That dumps $\log(P)$ or 'energy' into the *step_acceptor*'s degrees of freedom and raises its temperature. This parameter lets you define an approximate maximum temperature. Setting this may speed up convergence, though there is a corresponding risk of getting trapped in a local minimum.
(*type=float or None*)

Overrides: *object.__init__*

`T(self)`

Return Value

the current effective temperature.
(*type=float*)

__call__(*self*, *delta*)
Parameters**delta**: How much did the candidate step change $\log(P)$?*(type=float)***Return Value**

Whether to accept the candidate step or not.

(type=bool)

__repr__(*self*)
repr(*x*)

Overrides: object.__repr__ exitit(inherited documentation)

is_root(*self*)

This is a stub for compatibility with MPI code.

size(*self*)

This is a stub for compatibility with MPI code.

rank(*self*)

This is a stub for compatibility with MPI code.

Inherited from object

__delattr__(), __format__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
 __setattr__(), __sizeof__(), __str__(), __subclasshook__()

79.7.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

80 Module *gmisclib.mcmc_idxr*

80.1 Functions

logp_prior_normalized(*x*, *guess_prob_dist*)

Parameters

guess_prob_dist: (*type=list((str, probdist_c))*)

x: (*type=newstem2.indexclass.index_base*)

Return Value

the log of the prior probability.

(*type=float*)

80.2 Variables

Name	Description
<code>__package__</code>	Value: 'gmisclib'

80.3 Class *probdist_c*

object └─ *gmisclib.mcmc_idxr.probdist_c*

Known Subclasses: *gmisclib.mcmc_idxr.Expo*, *gmisclib.mcmc_idxr.LogNormal*, *gmisclib.mcmc_idxr.Normal*, *gmisclib.mcmc_idxr.Uniform*, *gmisclib.mcmc_idxr.Weibull*

80.3.1 Methods

__call__(*self*)

Sample from the probability distribution.

Return Value

the sample

(*type=float*)

logdens (<i>self</i> , <i>x</i>)
Return the log(density) of the probability distribution at <i>x</i> .
Parameters
<i>x</i> : the position where the density should be evaluated. (<i>type=</i> <i>float</i>)
Return Value
log(density) (<i>type=</i> <i>float</i>)
Raises
mcmc.NotGoodPosition when the density is not defined at <i>x</i> .

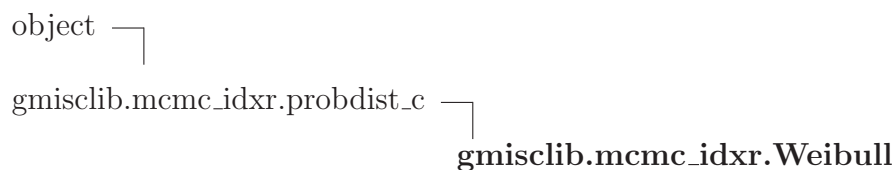
Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__init__()`, `__new__()`, `__reduce__()`,
`__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

80.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

80.4 Class Weibull



80.4.1 Methods

__init__ (<i>self</i> , <i>scale</i> , <i>shape</i>)
<i>x</i> . <code>__init__</code> (...) initializes <i>x</i> ; see <i>x</i> . <code>__class__</code> . <code>__doc__</code> for signature
Overrides: <code>object.__init__</code> extit(inherited documentation)

__call__(*self*)

Sample from the probability distribution.

Return Value

the sample

(*type=float*)

Overrides: gmisclib.mcmc_idxr.probdist_c.__call__ extit(inherited documentation)

logdens(*self*, *x*)

Return the log(density) of the probability distribution at *x*.

Parameters

x: the position where the density should be evaluated.

Return Value

log(density)

(*type=float*)

Raises

mcmc.NotGoodPosition when the density is not defined at *x*.

Overrides: gmisclib.mcmc_idxr.probdist_c.logdens extit(inherited documentation)

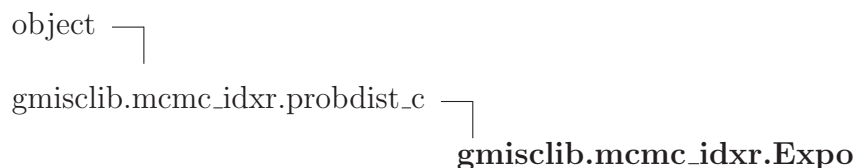
Inherited from object

__delattr__(), __format__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

80.4.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

80.5 Class Expo



80.5.1 Methods

__init__(*self*, *lmbda*)

x.__init__(...) initializes *x*; see *x*.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

__call__(*self*)

Sample from the probability distribution.

Return Value

the sample

(*type=float*)

Overrides: gmisclib.mcmc_idxr.probdist_c.__call__ extit(inherited documentation)

logdens(*self*, *x*)

Return the log(density) of the probability distribution at *x*.

Parameters

x: the position where the density should be evaluated.

Return Value

log(density)

(*type=float*)

Raises

`mcmc.NotGoodPosition` when the density is not defined at *x*.

Overrides: gmisclib.mcmc_idxr.probdist_c.logdens extit(inherited documentation)

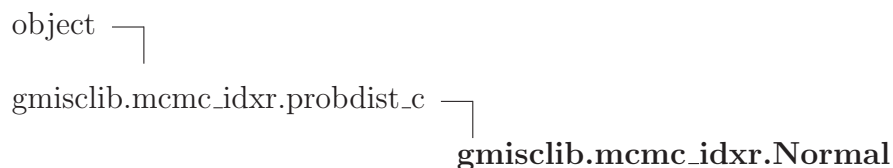
Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

80.5.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

80.6 Class Normal



80.6.1 Methods

`__init__(self, mu, sigma)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` `exitit`(inherited documentation)

`__call__(self)`

Sample from the probability distribution.

Return Value

the sample

(*type=float*)

Overrides: `gmisclib.mcmc_idxr.probdist_c.__call__` `exitit`(inherited documentation)

logdens(*self*, *x*)

Return the log(density) of the probability distribution at *x*.

Parameters

x: the position where the density should be evaluated.

Return Value

log(density)

(*type=float*)

Raises

`mcmc.NotGoodPosition` when the density is not defined at *x*.

Overrides: `gmisclib.mcmc_idxr.probdist_c.logdens` extit(inherited documentation)

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

80.6.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

80.7 Class Uniform

object └

gmisclib.mcmc_idxr.probdist_c └
 gmisclib.mcmc_idxr.Uniform

80.7.1 Methods

__init__(*self*, *low*, *high*)

x.`__init__`(...) initializes *x*; see *x*.`__class__`.`__doc__` for signature

Overrides: `object.__init__` extit(inherited documentation)

__call__(*self*)

Sample from the probability distribution.

Return Value

the sample

(*type=float*)

Overrides: gmisclib.mcmc_idxr.probdist_c.__call__ extit(inherited documentation)

logdens(*self*, *x*)

Return the log(density) of the probability distribution at *x*.

Parameters

x: the position where the density should be evaluated.

Return Value

log(density)

(*type=float*)

Raises

mcmc.NotGoodPosition when the density is not defined at *x*.

Overrides: gmisclib.mcmc_idxr.probdist_c.logdens extit(inherited documentation)

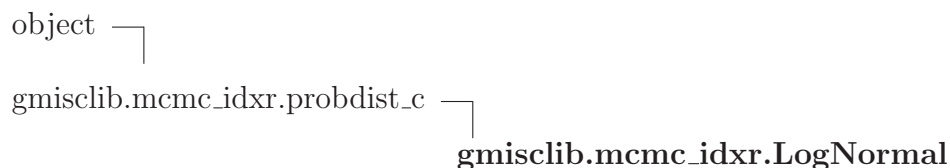
Inherited from object

__delattr__(), __format__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

80.7.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

80.8 Class *LogNormal*



80.8.1 Methods

`__init__(self, mu, sigma)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` `exitit`(inherited documentation)

`__call__(self)`

Sample from the probability distribution.

Return Value

the sample

(*type=*`float`)

Overrides: `gmisclib.mcmc_idxr.probdist_c.__call__` `exitit`(inherited documentation)

`logdens(self, x)`

Return the `log(density)` of the probability distribution at `x`.

Parameters

`x`: the position where the density should be evaluated.

Return Value

`log(density)`

(*type=*`float`)

Raises

`mcmc.NotGoodPosition` when the density is not defined at `x`.

Overrides: `gmisclib.mcmc_idxr.probdist_c.logdens` `exitit`(inherited documentation)

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

80.8.2 Properties

Name	Description
<i>Inherited from object</i> __class__	

81 Module gmisclib.mcmc_logger

81.1 Functions

read_multisample(*fname*, *Nsamp*=10, *tail*=0.0, *trigger*=None)

Read in a log file: it gives you multiple samples from the log. In other words, it provides a time-series of the changes to the parameters.

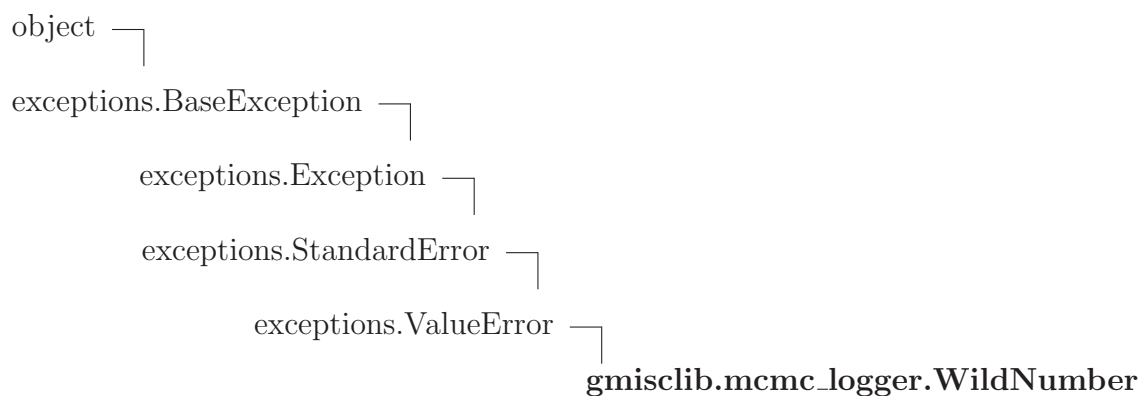
Parameters

tail: Where to start in the file? **tail**=0 means that you start at the trigger or beginning. **tail**=1-**epsilon** means you take just the last tiny bit of the file.

81.2 Variables

Name	Description
<code>--package--</code>	Value: 'gmisclib'

81.3 Class WildNumber



81.3.1 Methods

__init__(*self*, **s*)

x.**__init__**(...) initializes *x*; see *x*.**__class__**.**__doc__** for signature

Overrides: *object*.**__init__** extit(inherited documentation)

Inherited from exceptions.ValueError

`__new__()`

Inherited from `exceptions.BaseException`

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

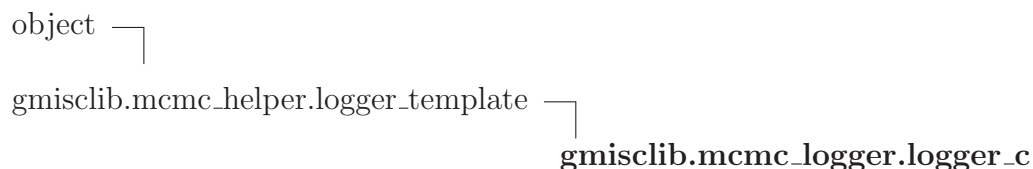
Inherited from `object`

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

81.3.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
<code>args</code> , <code>message</code>	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

81.4 Class `logger_c`



Known Subclasses: `gmislib.mcmc_logger.logger_A`, `gmislib.mcmc_logger.logger_N`

This class is called from the Markov-Chain stepper and writes the log file.

81.4.1 Methods

`__init__(self, logwriter, iterstep=100, ergstep=0.1)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` `exitit` (inherited documentation)

`set_logstep(self, iterstep, ergstep, reason='sampled')`

add (<i>self</i> , <i>stepperInstance</i> , <i>iter</i>)

Called every step: this decides whether data should be logged.
--

Overrides: <code>gmisclib.mcmc_helper.logger_template.add</code>
--

Add (<i>self</i> , <i>stepperInstance</i> , <i>iter</i> , <i>reason=None</i>)
--

close (<i>self</i>)

Overrides: <code>gmisclib.mcmc_helper.logger_template.close</code>
--

comment (<i>self</i> , <i>comment</i>)

header (<i>self</i> , <i>k</i> , <i>v</i>)

headers (<i>self</i> , <i>hdict</i>)

reset (<i>self</i>)

Overrides: <code>gmisclib.mcmc_helper.logger_template.reset</code>
--

set_trigger_point (<i>self</i>)
--

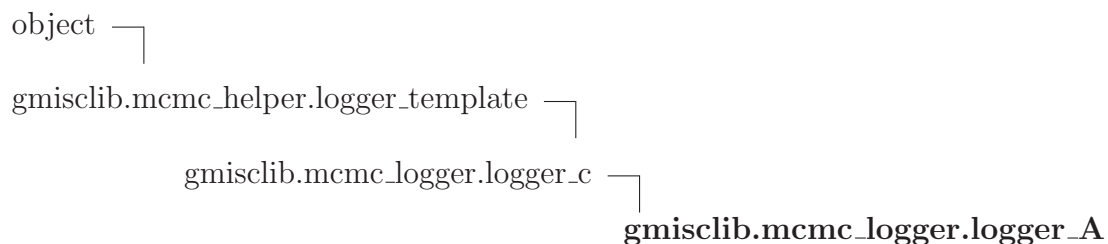
Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

81.4.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

81.5 Class *logger_A*



81.5.1 Methods

`__init__(self, logwriter, iterstep=100, ergstep=0.1)`

logwriter is a `fiatio.writer` or `avio.writer` or similar instance.

Overrides: `object.__init__`

`ok_logp(self, logpval)`

`ok_prms(self, prms)`

`fmt(self, prmname)`

`get_map(self, stepper)`

`Add(self, stepperInstance, iter, reason=None)`

Overrides: `gmisclib.mcmc_logger.logger_c.Add`

`add(self, stepperInstance, iter)`

Called every step: this decides whether data should be logged.

Overrides: `gmisclib.mcmc_helper.logger_template.add`

`close(self)`

Overrides: `gmisclib.mcmc_helper.logger_template.close`

`comment(self, comment)`

`header(self, k, v)`

<code>headers(self, hdict)</code>

<code>reset(self)</code>

Overrides: <code>gmisclib.mcmc_helper.logger_template.reset</code>
--

<code>set_logstep(self, iterstep, ergstep, reason='sampled')</code>

<code>set_trigger_point(self)</code>

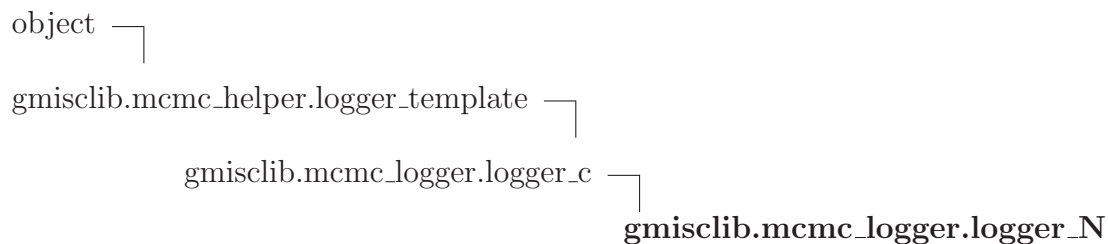
Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

81.5.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

81.6 Class `logger_N`



81.6.1 Methods

<code>__init__(self, logwriter, iterstep=100, ergstep=0.1)</code>

<code>logwriter</code> is really a <code>newstem2.newlogger</code> instance.
--

Overrides: <code>object.__init__</code>

Add (<i>self</i> , <i>stepperInstance</i> , <i>iter</i> , <i>reason</i> =None)
--

Actually write a set of parameters into the log file.

Overrides: <i>gmisclib.mcmc_logger.logger_c.Add</i>

add (<i>self</i> , <i>stepperInstance</i> , <i>iter</i>)

Called every step: this decides whether data should be logged.
--

Overrides: <i>gmisclib.mcmc_helper.logger_template.add</i>
--

close (<i>self</i>)

Overrides: <i>gmisclib.mcmc_helper.logger_template.close</i>
--

comment (<i>self</i> , <i>comment</i>)

header (<i>self</i> , <i>k</i> , <i>v</i>)

headers (<i>self</i> , <i>hdict</i>)

reset (<i>self</i>)

Overrides: <i>gmisclib.mcmc_helper.logger_template.reset</i>
--

set_logstep (<i>self</i> , <i>iterstep</i> , <i>ergstep</i> , <i>reason</i> ='sampled')

set_trigger_point (<i>self</i>)
--

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

81.6.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

81.7 Class `lti_c`



81.7.1 Methods

`__init__(self)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` `extit`(inherited documentation)

`parse(self, d)`

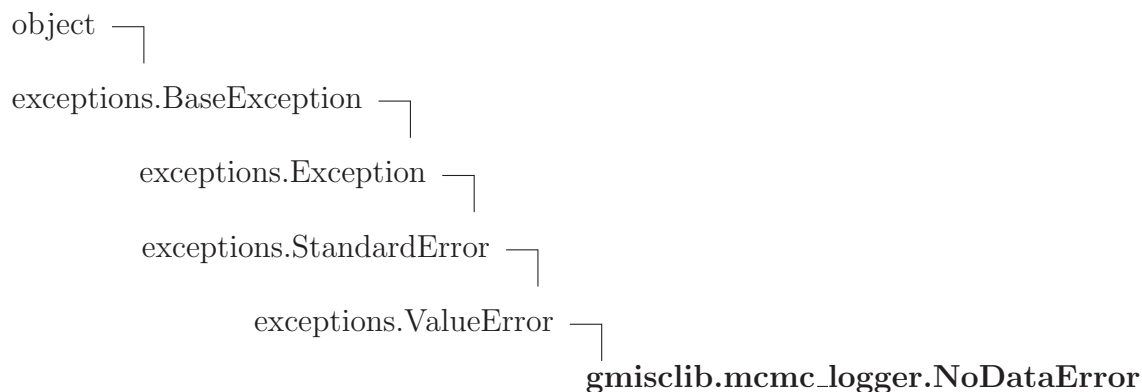
Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

81.7.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

81.8 Class `NoDataError`



81.8.1 Methods

`__init__`(*self*, **s*)

x.**`__init__`**(...) initializes *x*; see *x*.**`__class__`**.**`__doc__`** for signature

Overrides: *object*.**`__init__`** *exitit*(inherited documentation)

Inherited from exceptions.ValueError

`__new__`()

Inherited from exceptions.BaseException

`__delattr__`(), **`__getattr__`**(), **`__getitem__`**(), **`__getslice__`**(), **`__reduce__`**(), **`__repr__`**(),
`__setattr__`(), **`__setstate__`**(), **`__str__`**(), **`__unicode__`**()

Inherited from object

`__format__`(), **`__hash__`**(), **`__reduce_ex__`**(), **`__sizeof__`**(), **`__subclasshook__`**()

81.8.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
<i>args</i> , <i>message</i>	
<i>Inherited from object</i>	
<code>__class__</code>	

82 Module *gmisclib.mcmc_m4p*

This is a helper module to make use of *mcmc.py* and *mcmc_big.py*. It allows you to conveniently run a Monte-Carlo simulation of any kind until it converges (*stepper.run_to_bottom*) or until it has explored a large chunk of parameter space (*stepper.run_to_ergodic*).

It also helps you with logging the process.

When run in parallel, each processor does its thing more-or-less independently. However, every few steps, they exchange notes on their current progress. If one finds an improved vertex, it will be passed on to other processors via MPI.

Tested with *mpi4py* version 1.1.0, <http://mpi4py.scipy.org>

82.1 Functions

precompute_logp (<i>lop</i>)

Does a parallel evaluation of logp for all items in <i>lop</i> .
--

is_root ()

size ()

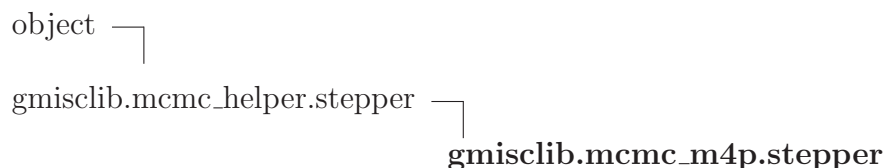
rank ()

test ()

82.2 Variables

Name	Description
Debug	Value: 0
mpi	Value: <code>MPI.COMM_WORLD</code>

82.3 Class stepper



82.3.1 Methods

__init__(*self*, *x*, *maxloops*=-1, *logger*=None)

x.__init__(...) initializes *x*; see *x*.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

reset_loops(*self*, *maxloops*=-1)

Overrides: gmisclib.mcmc_helper.stepper.reset_loops

communicate_hook(*self*, *id*)

Overrides: gmisclib.mcmc_helper.stepper.communicate_hook

join(*self*, *id*)

note(*self*, *s*, *lvl*)

Overrides: gmisclib.mcmc_helper.stepper.note

close(*self*)

After calling close(), it is no longer legal to call run_to_change(), run_to_ergodic(), or run_to_bottom(). Logging is shut down, but the contents of the stepper are still available for inspection.

run_to_bottom(*self*, *ns*=3, *acceptable_step*=None)

Run the *Markov Chain Monte-Carlo* until it converges near a minimum.

The general idea of the termination condition is that it keeps track of the angle between successive sequences of steps, where each sequence contains **ns** steps. If the angle is less than 90 degrees, it is evidence that the solution is still drifting in some consistent direction and therefore not yet at the maximum. Angles of greater than 90 degrees suggest that the optimization is wandering aimlessly near a minimum. It will run until a sufficient number of large angles are seen since the last BMCMC reset. A similar check is made on individual components of the parameter vector.

Parameters

ns: This controls how carefully it checks that it has really found a minimum. Large values will take longer but will assure more accurate convergence.

(*type*=**int**)

acceptable_step: A callable object (i.e. class or function) that returns **True** or **False**, depending on whether a proposed step is acceptable. See `mcmc.T_acceptor` for an example.

Raises

`TooManyLoops` ?

run_to_change(*self*, *ncw*=1, *acceptable_step*=None, *update_T*=False)

Run the *Markov Chain Monte-Carlo* until it finds an acceptable step.

Parameters

- ncw:** How many steps should be accepted before it stops?
(*type*=int)
- acceptable_step:** A function that decides what steps are acceptable and what are not. This is passed to the MCMC stepper, `mcmc.BootStepper`.
(*type*=function(float) -> bool, often *mcmc.T_acceptor* or *step_acceptor*)
- update_T:** Obsolete. Must be False.
(*type*=boolean.)

Raises

- TooManyLoops** if it takes a long time before enough steps are accepted.

run_to_ergodic(*self*, *ncw*=1, *T*=1.0)

Run the stepper until it has explored all of parameter space **ncw** times (as best as we can estimate). Note that this is a pretty careful routine. If the stepper finds a better region of parameter space so that log(P) improves part way through the process, the routine will reset itself and begin again.

NOTE: this sets **T** and **sortstrategy** for the **stepper**.

Parameters

- ncw:** how many times (or what fraction) of an ergodic exploration to make.
(*type*=int.)
- T:** temperature at which to run the Monte-Carlo process. This is normally 1.0. If it is **None**, then the current temperature is not modified.
(*type*=float or None.)

Return Value

- a **position** at the end of the process.

synchronize_abort(*self*, *id*)

synchronize_end(*self*, *id*)

<code>synchronize_start(self, id)</code>
--

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

82.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

82.3.3 Class Variables

Name	Description
<code>MPIID</code>	Value: 1241

83 Module `gmisclib.mcmc_mpi`

This is a helper module to make use of `mcmc.py` and `mcmc_big.py`. It allows you to conveniently run a Monte-Carlo simulation of any kind until it converges (`stepper.run_to_bottom`) or until it has explored a large chunk of parameter space (`stepper.run_to_ergodic`).

It also helps you with logging the process.

When run in parallel, each processor does its thing more-or-less independently. However, every few steps, they exchange notes on their current progress. If one finds an improved vertex, it will be passed on to other processors via MPI.

83.1 Functions

<code>precompute_logp(logp)</code>

Does a parallel evaluation of <code>logp</code> for all items in <code>logp</code> .
--

<code>test()</code>

83.2 Variables

Name	Description
Debug	Value: 0

83.3 Class `stepper`

object └

`gmisclib.mcmc_helper.stepper` └

`gmisclib.mcmc_mpi.stepper`

83.3.1 Methods

<code>__init__(self, x, maxloops=-1, logger=None, share=None)</code>

<code>x.__init__(...)</code> initializes <code>x</code> ; see <code>x.__class__.__doc__</code> for signature
--

Overrides: <code>object.__init__</code> <code>exitit</code> (inherited documentation)

reset_loops(*self*, *maxloops*=-1)Overrides: *gmisclib.mcmc_helper.stepper.reset_loops***communicate_hook**(*self*, *id*)Overrides: *gmisclib.mcmc_helper.stepper.communicate_hook***synchronize_start**(*self*, *id*)Overrides: *gmisclib.mcmc_helper.stepper.synchronize_start***synchronize_end**(*self*, *id*)Overrides: *gmisclib.mcmc_helper.stepper.synchronize_end***synchronize_abort**(*self*, *id*)Overrides: *gmisclib.mcmc_helper.stepper.synchronize_abort***synchronize**(*self*, *id*)**note**(*self*, *s*, *lvl*)Overrides: *gmisclib.mcmc_helper.stepper.note***size**(*self*)**rank**(*self*)**close**(*self*)

After calling `close()`, it is no longer legal to call `run_to_change()`, `run_to_ergodic()`, or `run_to_bottom()`. Logging is shut down, but the contents of the stepper are still available for inspection.

run_to_bottom(*self*, *ns*=3, *acceptable_step*=None)

Run the *Markov Chain Monte-Carlo* until it converges near a minimum.

The general idea of the termination condition is that it keeps track of the angle between successive sequences of steps, where each sequence contains **ns** steps. If the angle is less than 90 degrees, it is evidence that the solution is still drifting in some consistent direction and therefore not yet at the maximum. Angles of greater than 90 degrees suggest that the optimization is wandering aimlessly near a minimum. It will run until a sufficient number of large angles are seen since the last BMCMC reset. A similar check is made on individual components of the parameter vector.

Parameters

ns: This controls how carefully it checks that it has really found a minimum. Large values will take longer but will assure more accurate convergence.

(*type*=**int**)

acceptable_step: A callable object (i.e. class or function) that returns **True** or **False**, depending on whether a proposed step is acceptable. See `mcmc.T_acceptor` for an example.

Raises

`TooManyLoops` ?

run_to_change(*self*, *ncw*=1, *acceptable_step*=None, *update_T*=False)

Run the *Markov Chain Monte-Carlo* until it finds an acceptable step.

Parameters

ncw: How many steps should be accepted before it stops?
(*type=int*)

acceptable_step: A function that decides what steps are acceptable and what are not. This is passed to the MCMC stepper, `mcmc.BootStepper`.
(*type=function(float) -> bool, often `mcmc.T_acceptor` or `step_acceptor`*)

update_T: Obsolete. Must be False.
(*type=boolean.*)

Raises

TooManyLoops if it takes a long time before enough steps are accepted.

run_to_ergodic(*self*, *ncw*=1, *T*=1.0)

Run the stepper until it has explored all of parameter space **ncw** times (as best as we can estimate). Note that this is a pretty careful routine. If the stepper finds a better region of parameter space so that log(P) improves part way through the process, the routine will reset itself and begin again.

NOTE: this sets **T** and **sortstrategy** for the **stepper**.

Parameters

ncw: how many times (or what fraction) of an ergodic exploration to make.
(*type=int.*)

T: temperature at which to run the Monte-Carlo process. This is normally 1.0. If it is **None**, then the current temperature is not modified.
(*type=float or None.*)

Return Value

a **position** at the end of the process.

Inherited from object

`--delattr--()`, `--format--()`, `--getattr--()`, `--hash--()`, `--new--()`, `--reduce--()`, `--reduce_ex--()`, `--repr--()`, `--setattr--()`, `--sizeof--()`, `--str--()`, `--subclasshook--()`

83.3.2 Properties

Name	Description
<i>Inherited from object</i> __class__	

83.3.3 Class Variables

Name	Description
MPIID	Value: 1241

84 Module `gmisclib.mcmc_pypar`

This is a helper module to make use of `mcmc.py` and `mcmc_big.py`. It allows you to conveniently run a Monte-Carlo simulation of any kind until it converges (`stepper.run_to_bottom`) or until it has explored a large chunk of parameter space (`stepper.run_to_ergodic`).

It also helps you with logging the process.

84.1 Functions

Reduce (<i>x</i> , <i>root</i>)
--

precompute_logp (<i>lop</i>)

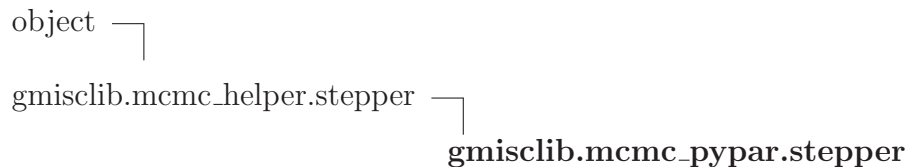
Does a parallel evaluation of logp for all items in <i>lop</i> .
--

is_root ()

size ()

test ()

84.2 Class `stepper`



84.2.1 Methods

__init__ (<i>self</i> , <i>x</i> , <i>maxloops</i> =-1, <i>logger</i> =None)
--

<i>x</i> . __init__ (...) initializes <i>x</i> ; see <i>x</i> . __class__ . __doc__ for signature
--

Overrides: <i>object</i> . __init__ <i>exitit</i> (inherited documentation)
--

reset_loops (<i>self</i> , <i>maxloops</i> =-1)

Overrides: <i>gmisclib.mcmc_helper.stepper</i> . reset_loops

communicate_hook(*self*)

Overrides: `gmisclib.mcmc_helper.stepper.communicate_hook`

join(*self*, *id*)

note(*self*, *s*)

Overrides: `gmisclib.mcmc_helper.stepper.note`

close(*self*)

After calling `close()`, it is no longer legal to call `run_to_change()`, `run_to_ergodic()`, or `run_to_bottom()`. Logging is shut down, but the contents of the stepper are still available for inspection.

run_to_bottom(*self*, *ns*=3, *acceptable_step*=None)

Run the *Markov Chain Monte-Carlo* until it converges near a minimum.

The general idea of the termination condition is that it keeps track of the angle between successive sequences of steps, where each sequence contains **ns** steps. If the angle is less than 90 degrees, it is evidence that the solution is still drifting in some consistent direction and therefore not yet at the maximum. Angles of greater than 90 degrees suggest that the optimization is wandering aimlessly near a minimum. It will run until a sufficient number of large angles are seen since the last BMCMC reset. A similar check is made on individual components of the parameter vector.

Parameters

ns: This controls how carefully it checks that it has really found a minimum. Large values will take longer but will assure more accurate convergence.

(*type*=*int*)

acceptable_step: A callable object (i.e. class or function) that returns **True** or **False**, depending on whether a proposed step is acceptable. See `mcmc.T_acceptor` for an example.

Raises

`TooManyLoops` ?

run_to_change(*self*, *ncw*=1, *acceptable_step*=None, *update_T*=False)

Run the *Markov Chain Monte-Carlo* until it finds an acceptable step.

Parameters

- ncw:** How many steps should be accepted before it stops?
(*type*=int)
- acceptable_step:** A function that decides what steps are acceptable and what are not. This is passed to the MCMC stepper, `mcmc.BootStepper`.
(*type*=function(float) -> bool, often *mcmc.T_acceptor* or *step_acceptor*)
- update_T:** Obsolete. Must be False.
(*type*=boolean.)

Raises

- TooManyLoops** if it takes a long time before enough steps are accepted.

run_to_ergodic(*self*, *ncw*=1, *T*=1.0)

Run the stepper until it has explored all of parameter space **ncw** times (as best as we can estimate). Note that this is a pretty careful routine. If the stepper finds a better region of parameter space so that log(P) improves part way through the process, the routine will reset itself and begin again.

NOTE: this sets **T** and **sortstrategy** for the **stepper**.

Parameters

- ncw:** how many times (or what fraction) of an ergodic exploration to make.
(*type*=int.)
- T:** temperature at which to run the Monte-Carlo process. This is normally 1.0. If it is **None**, then the current temperature is not modified.
(*type*=float or None.)

Return Value

- a **position** at the end of the process.

synchronize_abort(*self*, *id*)

synchronize_end(*self*, *id*)

<code>synchronize_start(self, id)</code>
--

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

84.2.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

84.2.3 Class Variables

Name	Description
<code>MPIID</code>	Value: 1241

85 Module `gmisclib.mcmc_socket`

This is a helper module to make use of `mcmc.py` and `mcmc_big.py`. It allows you to conveniently run a Monte-Carlo simulation of any kind until it converges (`stepper.run_to_bottom`) or until it has explored a large chunk of parameter space (`stepper.run_to_ergodic`).

It also helps you with logging the process.

When run in parallel, each processor does its thing more-or-less independently. However, every few steps, they exchange notes on their current progress. If one finds an improved vertex, it will be passed on to other processors via `mcmc.cooperate.py`.

85.1 Functions

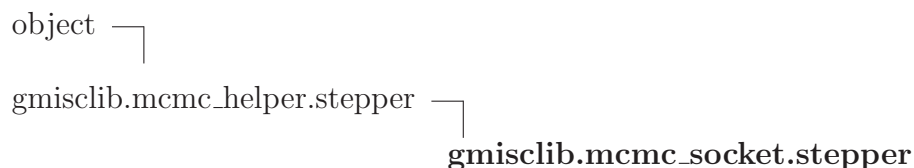
```
precompute_logp(lop)
```

```
test(args)
```

85.2 Variables

Name	Description
Debug	Value: 0
__package__	Value: 'gmisclib'

85.3 Class `stepper`



85.3.1 Methods

```
__init__(self, x, maxloops=-1, logger=None, share=None)
x.__init__(...) initializes x; see x.__class__.__doc__ for signature
Overrides: object.__init__ extit(inherited documentation)
```

reset_loops(*self*, *maxloops*=-1)Overrides: *gmisc.lib.mcmc.helper.stepper.reset_loops***communicate_hook**(*self*, *id*)Overrides: *gmisc.lib.mcmc.helper.stepper.communicate_hook***synchronize_start**(*self*, *id*)Overrides: *gmisc.lib.mcmc.helper.stepper.synchronize_start***synchronize_end**(*self*, *id*)Overrides: *gmisc.lib.mcmc.helper.stepper.synchronize_end***synchronize_abort**(*self*, *id*)Overrides: *gmisc.lib.mcmc.helper.stepper.synchronize_abort***note**(*self*, *s*, *lvl*)Overrides: *gmisc.lib.mcmc.helper.stepper.note***close**(*self*)

After calling `close()`, it is no longer legal to call `run_to_change()`, `run_to_ergodic()`, or `run_to_bottom()`. Logging is shut down, but the contents of the stepper are still available for inspection.

Overrides: *gmisc.lib.mcmc.helper.stepper.close* `exitit`(inherited documentation)**size**(*self*)**rank**(*self*)

run_to_bottom(*self*, *ns*=3, *acceptable_step*=None)

Run the *Markov Chain Monte-Carlo* until it converges near a minimum.

The general idea of the termination condition is that it keeps track of the angle between successive sequences of steps, where each sequence contains **ns** steps. If the angle is less than 90 degrees, it is evidence that the solution is still drifting in some consistent direction and therefore not yet at the maximum. Angles of greater than 90 degrees suggest that the optimization is wandering aimlessly near a minimum. It will run until a sufficient number of large angles are seen since the last BMCMC reset. A similar check is made on individual components of the parameter vector.

Parameters

ns: This controls how carefully it checks that it has really found a minimum. Large values will take longer but will assure more accurate convergence.

(*type*=**int**)

acceptable_step: A callable object (i.e. class or function) that returns **True** or **False**, depending on whether a proposed step is acceptable. See `mcmc.T_acceptor` for an example.

Raises

`TooManyLoops` ?

run_to_change(*self*, *ncw*=1, *acceptable_step*=None, *update_T*=False)

Run the *Markov Chain Monte-Carlo* until it finds an acceptable step.

Parameters

ncw: How many steps should be accepted before it stops?
(*type=int*)

acceptable_step: A function that decides what steps are acceptable and what are not. This is passed to the MCMC stepper, `mcmc.BootStepper`.
(*type=function(float) -> bool, often `mcmc.T_acceptor` or `step_acceptor`*)

update_T: Obsolete. Must be False.
(*type=boolean.*)

Raises

TooManyLoops if it takes a long time before enough steps are accepted.

run_to_ergodic(*self*, *ncw*=1, *T*=1.0)

Run the stepper until it has explored all of parameter space **ncw** times (as best as we can estimate). Note that this is a pretty careful routine. If the stepper finds a better region of parameter space so that log(P) improves part way through the process, the routine will reset itself and begin again.

NOTE: this sets **T** and **sortstrategy** for the **stepper**.

Parameters

ncw: how many times (or what fraction) of an ergodic exploration to make.
(*type=int.*)

T: temperature at which to run the Monte-Carlo process. This is normally 1.0. If it is **None**, then the current temperature is not modified.
(*type=float or None.*)

Return Value

a **position** at the end of the process.

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

85.3.2 Properties

Name	Description
<i>Inherited from object</i> __class__	

86 Module `gmisclib.md_scaling`

Find a set of points so that the distances between them match a desired set of distances.

86.1 Functions

```
find_rep(dij, ndim)
```

```
distances(xi)
```

```
test()
```

86.2 Variables

Name	Description
EPS	Value: 1e-3
G	Value: 1

87 Module *gmisclib.multivariate_classes*

Support module for *multivariate.py*

87.1 Functions

`vec_inv_variance(start)`

`diag_inv_variance(start)`

87.2 Variables

Name	Description
BAYES_PRIOR_SPACE	Value: 0
HUGE	Value: 1e+30
<code>__package__</code>	Value: 'gmisclib'

87.3 Class *QuadraticNotNormalizable*



87.3.1 Methods

`__init__(self, s='')`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` `extit`(inherited documentation)

Inherited from `exceptions.ValueError`

`__new__()`

Inherited from exceptions.BaseException

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

Inherited from object

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

87.3.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
args, message	
<i>Inherited from object</i>	
__class__	

87.4 Class modeldesc

Known Subclasses: gmsclib.multivariate_mm.multi_mu, gmsclib.multivariate_mm.multi_mu_diag, gmsclib.multivariate_q.quadratic

Virtual base class for a description of a model of a particular size.

87.4.1 Methods

`__init__(self, ndim)`

`ndim(self)`

This returns the dimensionality of the data.

`modeldim(self)`

This gives the dimensionality of the model, i.e. the number of parameters required to specify the means and covariance matrix(ces).

`unpack(self, prms)`

This returns some subclass of model.

new (<i>self</i> , <i>mu</i> , <i>invsigma</i>)
--

Creates a model that contains data.

start (<i>self</i> , <i>dataset</i>)

Selects a random starting point from the dataset.

87.4.2 Class Variables

Name	Description
<code>__doc__</code>	Value: ""Virtual base class for a description of a mo...
<code>LF</code>	Value: 0.111111111111

87.5 Class `model_with_numbers`

Known Subclasses: `gmisclib.multivariate_mm.multi_mu_diag_with_numbers`, `gmisclib.multivariate_mm.multi_mu_diag_without_numbers`, `gmisclib.multivariate_q.diag_quadratic_with_numbers`, `gmisclib.multivariate_q.quadratic_with_numbers`

Virtual base class for adding in the functions you need when you create a model with known parameters (like `mu` and `sigma`).

87.5.1 Methods

__init__ (<i>self</i> , <i>details</i> , <i>bias</i>)
--

Bias is an overall shift of the log probability up or down. In a classifier, it is used to bias things toward one class or another.

logp (<i>self</i> , <i>datum</i>)
--

pack (<i>self</i>)

Returns a vector of parameters.

ndim (<i>self</i>)

unpack (<i>self</i> , <i>prms</i>)

new (<i>self</i> , <i>mu</i> , <i>invsigma</i>)
--

```
start(self, dataset)
```

```
offset(self)
```

```
addoff(self)
```

87.5.2 Class Variables

Name	Description
<code>__doc__</code>	Value: <code>"""Virtual base class for adding in the functi...</code>

88 Module `gmisclib.multivariance_mm`

This a helper module for `multivariance.py`

88.1 Functions

```
__multi_mu__init__(self, dataset=None, ndim=None, idmap=None,
details=None)
```

You either give it a complete dataset to look at, including class IDs, *or* the dimensionality of the data (`ndim`) and a map between classids and integers. This map can be obtained from `nice_hash`.

88.2 Variables

Name	Description
<code>__package__</code>	Value: <code>'gmisclib'</code>

88.3 Class `datum_c`

88.3.1 Methods

```
__init__(self, vector, classid)
```

88.4 Class `multi_mu`

`gmisclib.multivariance_classes.modeldesc` — `gmisclib.multivariance_mm.multi_mu`

This describes a quadratic model of a known size, with multiple means (one for each different class of data).

88.4.1 Methods

`__init__(self, dataset=None, ndim=None, idmap=None, details=None)`

You either give it a complete dataset to look at, including class IDs, *or* the dimensionality of the data (`ndim`) and a map between classids and integers. This map can be obtained from `nice_hash`.

Overrides: `gmisclib.multivariance_classes.modeldesc.__init__`

`modeldim(self)`

This gives the dimensionality of the model, i.e. the number of parameters required to specify the means and covariance matrix(ces).

Overrides: `gmisclib.multivariance_classes.modeldesc.modeldim` `exitit`(inherited documentation)

`unpack(self, prms)`

This returns some subclass of model.

Overrides: `gmisclib.multivariance_classes.modeldesc.unpack` `exitit`(inherited documentation)

`new(self, mu, invsigma, bias=0.0)`

`Mu` is a mapping of classids to vectors. `invsigma` is a square matrix.

Overrides: `gmisclib.multivariance_classes.modeldesc.new`

`start(self, data)`

Selects a random starting point from the dataset.

Overrides: `gmisclib.multivariance_classes.modeldesc.start` `exitit`(inherited documentation)

`ndim(self)`

This returns the dimensionality of the data.

88.4.2 Class Variables

Name	Description
<code>__doc__</code>	Value: <code>""</code> "This describes a quadratic model of a known si..."

continued on next page

Name	Description
LF	Value: 0.111111111111

88.5 Class `multi_mu_with_numbers`

`gmisclib.multivariate_classes.model_with_numbers` — `gmisclib.multivariate_mm.multi_mu_with_numbers`

88.5.1 Methods

`__init__`(*self*, *mu*, *invsigma*, *details*, *bias*=0.0, *offset*=None)

self.mu, *self.invsigma*, and *self._offset* should be considered read-only for all users of this class.

Overrides: `gmisclib.multivariate_classes.model_with_numbers.__init__`

`__str__`(*self*)

`__repr__`(*self*)

`addoff`(*self*)

Overrides: `gmisclib.multivariate_classes.model_with_numbers.addoff`

`pack`(*self*)

Returns a vector of parameters.

Overrides: `gmisclib.multivariate_classes.model_with_numbers.pack`
`extit`(inherited documentation)

`logp`(*self*, *datum*)

Overrides: `gmisclib.multivariate_classes.model_with_numbers.logp`

`ndim`(*self*)

`new`(*self*, *mu*, *invsigma*)

`offset`(*self*)

<code>start(self, dataset)</code>

<code>unpack(self, prms)</code>

88.5.2 Class Variables

Name	Description
<code>__doc__</code>	Value: <code>""</code> "Virtual base class for adding in the functi..."

88.6 Class `multi_mu_diag`

`gmisclib.multivariate_classes.modeldesc` — `gmisclib.multivariate_mm.multi_mu_diag`

This describes a quadratic model of a known size, with multiple means (one for each different class of data). The covariance matrix is shared and diagonal.

88.6.1 Methods

<code>__init__(self, dataset=None, ndim=None, idmap=None, details=None)</code>
--

You either give it a complete dataset to look at, including class IDs, *or* the dimensionality of the data (`ndim`) and a map between classids and integers. This map can be obtained from `nice_hash`.

Overrides: `gmisclib.multivariate_classes.modeldesc.__init__`

<code>modeldim(self)</code>

This gives the dimensionality of the model, i.e. the number of parameters required to specify the means and covariance matrix(ces).

Overrides: `gmisclib.multivariate_classes.modeldesc.modeldim` `extit`(inherited documentation)

<code>unpack(self, prms)</code>

This returns some subclass of `model`.

Overrides: `gmisclib.multivariate_classes.modeldesc.unpack` `extit`(inherited documentation)

new(*self*, *mu*, *invsigma*, *bias*=0.0)

Mu is a mapping of classids to vectors. Invsigma is a vector.

Overrides: `gmisclib.multivariate_classes.modeldesc.new`

start(*self*, *data*)

Selects a random starting point from the dataset.

Overrides: `gmisclib.multivariate_classes.modeldesc.start` `exitit`(inherited documentation)

ndim(*self*)

This returns the dimensionality of the data.

88.6.2 Class Variables

Name	Description
<code>__doc__</code>	Value: <code>""</code> "This describes a quadratic model of a known si...
<code>LF</code>	Value: 0.111111111111

88.7 Class `multi_mu_diag_with_numbers`

`gmisclib.multivariate_classes.model_with_numbers`

`gmisclib.multivariate_mm.multi_mu_diag_wi`

88.7.1 Methods

__init__(*self*, *mu*, *invsigma*, *details*, *bias*=0.0, *offset*=None)

`self.mu`, `self.invsigma`, and `self._offset` should be considered read-only for all users of this class.

Overrides: `gmisclib.multivariate_classes.model_with_numbers.__init__`

__str__(*self*)

__repr__(*self*)

addoff(*self*)Overrides: *gmisclib.multivariate_classes.model_with_numbers.addoff***pack**(*self*)

Returns a vector of parameters.

Overrides: *gmisclib.multivariate_classes.model_with_numbers.pack*
extit(inherited documentation)**logp**(*self*, *datum*)Overrides: *gmisclib.multivariate_classes.model_with_numbers.logp***ndim**(*self*)**new**(*self*, *mu*, *invsigma*)**offset**(*self*)**start**(*self*, *dataset*)**unpack**(*self*, *prms*)

88.7.2 Class Variables

Name	Description
<code>__doc__</code>	Value: <code>"""Virtual base class for adding in the functi...</code>

89 Module `gmisclib.multivariate_q`

This a helper module for `multivariate.py`

89.1 Variables

Name	Description
<code>HUGE</code>	Value: <code>1e+30</code>
<code>__package__</code>	Value: <code>'gmisclib'</code>

89.2 Class `quadratic`

`gmisclib.multivariate_classes.modeldesc` — `gmisclib.multivariate_q.quadratic`

Known Subclasses: `gmisclib.multivariate_q.diag_quadratic`

This describes a quadratic model of a known size.

89.2.1 Methods

`__init__(self, dataset=None, ndim=None)`

Overrides: `gmisclib.multivariate_classes.modeldesc.__init__`

`modeldim(self)`

This gives the dimensionality of the model, i.e. the number of parameters required to specify the means and covariance matrix(es).

Overrides: `gmisclib.multivariate_classes.modeldesc.modeldim` `exitit`(inherited documentation)

`unpack(self, prms)`

This returns some subclass of model.

Overrides: `gmisclib.multivariate_classes.modeldesc.unpack` `exitit`(inherited documentation)

new(*self*, *mu*, *invsigma*, *bias*=0.0)

Creates a model that contains data.

Overrides: `gmisclib.multivariate_classes.modeldesc.new` extit(inherited documentation)

start(*self*, *data*)

Returns (starting position vector, covariance matrix)

Overrides: `gmisclib.multivariate_classes.modeldesc.start`

ndim(*self*)

This returns the dimensionality of the data.

89.2.2 Class Variables

Name	Description
<code>__doc__</code>	Value: <code>"""This describes a quadratic model of a known size."""</code>
<code>LF</code>	Value: <code>0.111111111111</code>

89.3 Class `quadratic_with_numbers`

`gmisclib.multivariate_classes.model_with_numbers`

└ `gmisclib.multivariate_q.quadratic_with_num`

89.3.1 Methods

__init__(*self*, *mu*, *invsigma*, *details*, *bias*=0.0, *offset*=None)

`self.mu`, `self.invsigma`, and `self._offset` should be considered read-only for all users of this class.

Overrides: `gmisclib.multivariate_classes.model_with_numbers.__init__`

__str__(*self*)

__repr__(*self*)

addoff(*self*)Overrides: *gmisclib.multivariate_classes.model_with_numbers.addoff***logp**(*self*, *datum*)Overrides: *gmisclib.multivariate_classes.model_with_numbers.logp***ndim**(*self*)**new**(*self*, *mu*, *invsigma*)**offset**(*self*)**pack**(*self*)

Returns a vector of parameters.

start(*self*, *dataset*)**unpack**(*self*, *prms*)

89.3.2 Class Variables

Name	Description
<code>__doc__</code>	Value: <code>"""Virtual base class for adding in the functi...</code>

89.4 Class *diag_quadratic*

*gmisclib.multivariate_classes.modeldesc**gmisclib.multivariate_q.quadratic**gmisclib.multivariate_q.diag_quadratic*

89.4.1 Methods

__init__(*self*, *dataset*=None, *ndim*=None)Overrides: *gmisclib.multivariate_classes.modeldesc.__init__*

modeldim(*self*)

This gives the dimensionality of the model, i.e. the number of parameters required to specify the means and covariance matrix(es).

Overrides: *gmisclib.multivariate_classes.modeldesc.modeldim* *exitit*(inherited documentation)

unpack(*self*, *prms*)

This returns some subclass of model.

Overrides: *gmisclib.multivariate_classes.modeldesc.unpack* *exitit*(inherited documentation)

new(*self*, *mu*, *invsigma*, *bias*=0.0)

Creates a model that contains data.

Overrides: *gmisclib.multivariate_classes.modeldesc.new* *exitit*(inherited documentation)

start(*self*, *data*)

Returns (starting position vector, covariance matrix)

Overrides: *gmisclib.multivariate_classes.modeldesc.start* *exitit*(inherited documentation)

ndim(*self*)

This returns the dimensionality of the data.

89.4.2 Class Variables

Name	Description
LF	Value: 0.111111111111
__doc__	Value: """This describes a quadratic model of a known size."""

89.5 Class `diag_quadratic_with_numbers`

`gmisclib.multivariate_classes.model_with_numbers`

`gmisclib.multivariate_q.diag_quadratic_with_numbers`

89.5.1 Methods

`__init__`(*self*, *mu*, *invsigma*, *details*, *bias*=0.0, *offset*=None)

Bias is an overall shift of the log probability up or down. In a classifier, it is used to bias things toward one class or another.

Overrides: `gmisclib.multivariate_classes.model_with_numbers.__init__`
`__init__`(inherited documentation)

`__str__`(*self*)

`__repr__`(*self*)

`addoff`(*self*)

Overrides: `gmisclib.multivariate_classes.model_with_numbers.addoff`

`logp`(*self*, *datum*)

Overrides: `gmisclib.multivariate_classes.model_with_numbers.logp`

`ndim`(*self*)

`new`(*self*, *mu*, *invsigma*)

`offset`(*self*)

`pack`(*self*)

Returns a vector of parameters.

`start`(*self*, *dataset*)

`unpack`(*self*, *prms*)

89.5.2 Class Variables

Name	Description
<code>__doc__</code>	Value: <code>"""Virtual base class for adding in the functi...</code>

90 Module *gmisc.lib.multivariate_normal*

90.1 Functions

<code>test()</code>

90.2 Variables

Name	Description
<code>--package--</code>	Value: <code>'gmisc.lib'</code>

90.3 Class *multivariate_normal*

`multivariate_normal(mean, cov)` or `multivariate_normal(mean, cov, [m, n, ...])` returns an array containing multivariate normally distributed random numbers with specified mean and covariance.

`mean` must be a 1 dimensional array. `cov` must be a square two dimensional array with the same number of rows and columns as `mean` has elements.

The first form returns a single 1-D array containing a multivariate normal.

The second form returns an array of shape `(m, n, ..., cov.shape[0])`. In this case, `output[i,j,...,:]` is a 1-D array containing a multivariate normal.

90.3.1 Methods

<code>--init--(self, mean, cov)</code>
--

<code>sample(self, shape=[])</code>

90.3.2 Class Variables

Name	Description
<code>--doc--</code>	Value: <code>""multivariate_normal(mean, cov) or multivariate_normal(...</code>

91 Module `gmisclib.named_block_file`

Read and write files in the form [label] text text text [label2] text text ...

The format is imperfect to the extent that it does not allow the text to contain things that look like labels.

When used as a script, it selects regions of such a file and prints them on the standard output: Usage: `named_block_file.py key1 [key2] <input >output` The sections labelled `key1` and `key2` will be output in order, with no separators other than a newline in between.

91.1 Functions

<code>read(<i>fd</i>)</code>

<code>write_key(<i>fd</i>, <i>k</i>)</code>

<code>write_text(<i>fd</i>, <i>t</i>)</code>
--

<code>write_line(<i>fd</i>, <i>t</i>)</code>
--

<code>write_line_kv(<i>fd</i>, <i>k</i>, <i>v</i>)</code>

<code>write(<i>fd</i>, <i>d</i>)</code>

<code>test()</code>

91.2 Variables

Name	Description
<code>--package--</code>	Value: 'gmisclib'

92 Module gmisclib.nbest

Beam search through a graph.

92.1 Functions

go(*graph*, *nbeam*, *cbeam*)

Search the graph for the lowest cost routes. Returns [(cost, route), ...] where cost is the cost of a route, and route is [node, node, node, ...] the list of nodes on the route. Routes are sorted in order of increasing cost.

test()

92.2 Variables

Name	Description
<code>--package--</code>	Value: None

92.3 Class node

This is a node of a graph, it includes links to other nodes.

92.3.1 Methods

--init--(*self*, *cost*=0.0, *label*=None, *terminal*=0, *comment*=None)

Create a node, with a specified cost (used in the beam search), and a label (arbitrary information). Terminal nodes are marked, and terminate the search.

add(*self*, *nextnode*, *cost*=0.0, *label*=None)

Add a link from self to nextnode. Links can have a cost and label, too.

--str--(*self*)

--repr--(*self*)

92.3.2 Class Variables

Name	Description
<code>--doc--</code>	Value: <code>""</code> "This is a node of a graph, it includes links to other ..."

93 Module gmisclib.nice_hash

93.1 Functions

```
map(list, trimmer=<function <lambda> at 0x24a0aa0>)
```

93.2 Variables

Name	Description
h	Value: nice_hash(lambda x: x)
__package__	Value: None

93.3 Class DontHashThis



If your trimmer function raises this exception, then the input to nice_hash.add() will be ignored.

93.3.1 Methods

```
__init__(self)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature
Overrides: object.__init__ extit(inherited documentation)
```

Inherited from exceptions.Exception

```
__new__()
```

Inherited from exceptions.BaseException

```
__delattr__(), __getattr__(), __getitem__(), __getslice__(), __reduce__(), __repr__(),
__setattr__(), __setstate__(), __str__(), __unicode__()
```

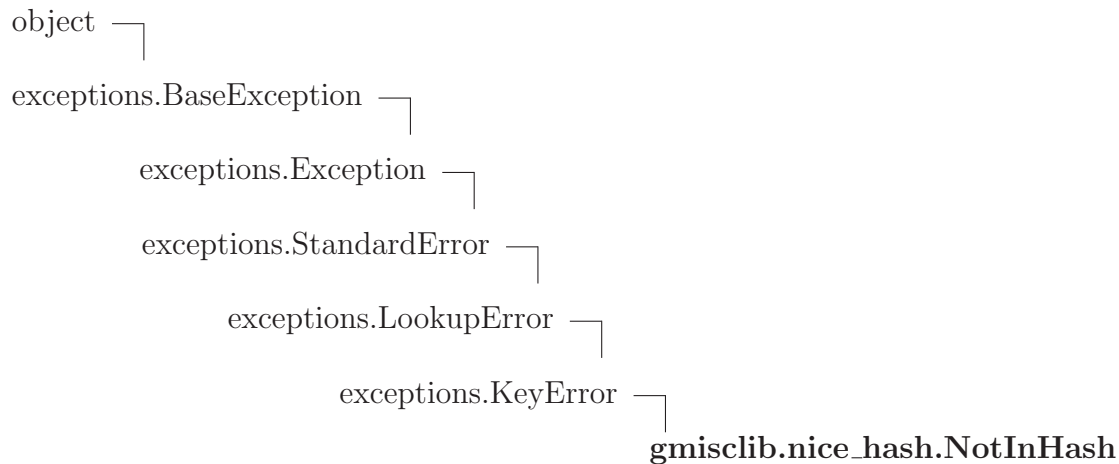
Inherited from object

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

93.3.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
<code>args</code> , <code>message</code>	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

93.4 Class *NotInHash*



93.4.1 Methods

```

__init__(self, s)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature
Overrides: object.__init__ extit(inherited documentation)

```

Inherited from `exceptions.KeyError`

`__new__()`, `__str__()`

Inherited from `exceptions.BaseException`

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__unicode__()`

Inherited from `object`

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

93.4.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
<code>args</code> , <code>message</code>	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

93.5 Class simple

object └─ **gmisclib.nice_hash.simple**

Known Subclasses: `gmisclib.nice_hash.hash`

This class generates a map from inputs to integers, where an equivalence class is defined by the trimmer function. It is just a stripped-down version of the `hash` class defined above. Faster and less memory consumption, if you don't need the `rmap` method.

93.5.1 Methods

<code>__init__(self, trimmer=None)</code>
<code>x.__init__(...)</code> initializes <code>x</code> ; see <code>x.__class__.__doc__</code> for signature
Parameters
<code>trimmer</code>: Defines equivalence classes on list; it can be some sort of projection operation. The default (<code>None</code>) makes the identity class equal to the inputs. It can also specify which inputs to ignore by raising the <code>DontHashThis</code> exception.
Overrides: <code>object.__init__</code>
<code>add(self, x)</code>
Returns an integer which is shared among all <code>x</code> in the equivalence class, but different from all other <code>x</code> . If the trimmer function raises the <code>DontHashThis</code> exception, it will ignore the input and return <code>None</code> .
Return Value
<code>int</code>

add_newonly(*self*, *x*)

Similar to add, except it returns None if the equivalence class has already appeared.

Return Value

int or None

get_image(*self*, *x*)

Returns an integer which is shared among all *x* in the equivalence class, but different from all other *x*.

map(*self*)

Return the map from classes (as returned by the trimmer function) to integers.

classes(*self*)

Return a list over all the classes that have been seen.

classmap(*self*)

Return the map from integers to classes.

Return Value

[classname, classname, ...]

Inherited from object

`__delattr__()`, `__format__()`, `__getattribute__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

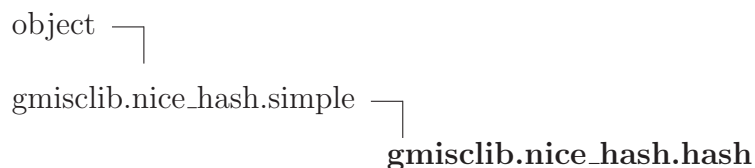
93.5.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

93.5.3 Class Variables

Name	Description
<code>__doc__</code>	Value: <code>""</code> "This class generates a map from inputs to intege..."

93.6 Class hash



93.6.1 Methods

__init__(*self*, *trimmer*=None)

This adds the **rmap** method to the **simple** class.

Parameters

trimmer: Defines equivalence classes on list; it can be some sort of projection operation. The default (None) makes the identity class equal to the inputs. It can also specify which inputs to ignore by raising the **DontHashThis** exception.

Overrides: **object.__init__**

rmap(*self*)

Returns the map from integers to inputs.

test(*self*)

add(*self*, *x*)

Returns an integer which is shared among all *x* in the equivalence class, but different from all other *x*. If the *trimmer* function raises the **DontHashThis** exception, it will ignore the input and return None. Note that if you add an item twice, it will appear twice in **rmap**().

Return Value

int

Overrides: **gmisclib.nice_hash.simple.add**

add_newonly(*self*, *x*)

Similar to add, except it returns None if the equivalence class has already appeared.

The list of classes seen (*self*.seen) then contains only the first item that was added in each class, so *rmap*() will produce a dictionary containing single-item lists: {int1:[item1],int2:[item2],int3:[item3], ...}. Calls to *add*() and *add_newonly*() can be intermixed safely, though the result of *rmap*() will then not contain all added items.

Return Value

int or None

Overrides: gmisclib.nice_hash.simple.add_newonly

classes(*self*)

Return a list over all the classes that have been seen.

classmap(*self*)

Return the map from integers to classes.

Return Value

[classname, classname, ...]

get_image(*self*, *x*)

Returns an integer which is shared among all *x* in the equivalence class, but different from all other *x*.

map(*self*)

Return the map from classes (as returned by the trimmer function) to integers.

Inherited from object

__delattr__(), *__format__*(), *__getattr__*(), *__hash__*(), *__new__*(), *__reduce__*(), *__reduce_ex__*(), *__repr__*(), *__setattr__*(), *__sizeof__*(), *__str__*(), *__subclasshook__*()

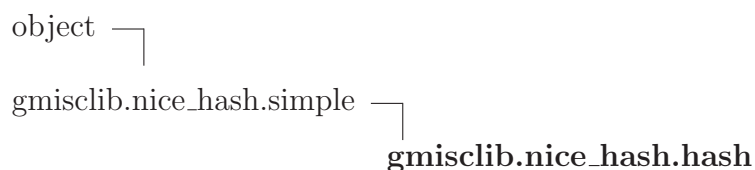
93.6.2 Properties

Name	Description
<i>Inherited from object</i>	
<i>__class__</i>	

93.6.3 Class Variables

Name	Description
<code>__doc__</code>	Value: <code>"""This class generates a map from inputs to intege...</code>

93.7 Class hash



93.7.1 Methods

`__init__(self, trimmer=None)`

This adds the `rmap` method to the `simple` class.

Parameters

trimmer: Defines equivalence classes on list; it can be some sort of projection operation. The default (None) makes the identity class equal to the inputs. It can also specify which inputs to ignore by raising the `DontHashThis` exception.

Overrides: `object.__init__`

`rmap(self)`

Returns the map from integers to inputs.

`test(self)`

`add(self, x)`

Returns an integer which is shared among all `x` in the equivalence class, but different from all other `x`. If the `trimmer` function raises the `DontHashThis` exception, it will ignore the input and return `None`. Note that if you add an item twice, it will appear twice in `rmap()`.

Return Value

`int`

Overrides: `gmisclib.nice_hash.simple.add`

add_newonly(*self*, *x*)

Similar to add, except it returns None if the equivalence class has already appeared.

The list of classes seen (*self*.seen) then contains only the first item that was added in each class, so *rmap*() will produce a dictionary containing single-item lists: {int1:[item1],int2:[item2],int3:[item3], ...}. Calls to *add*() and *add_newonly*() can be intermixed safely, though the result of *rmap*() will then not contain all added items.

Return Value

int or None

Overrides: gmisclib.nice_hash.simple.add_newonly

classes(*self*)

Return a list over all the classes that have been seen.

classmap(*self*)

Return the map from integers to classes.

Return Value

[classname, classname, ...]

get_image(*self*, *x*)

Returns an integer which is shared among all *x* in the equivalence class, but different from all other *x*.

map(*self*)

Return the map from classes (as returned by the trimmer function) to integers.

Inherited from object

__delattr__(), *__format__*(), *__getattr__*(), *__hash__*(), *__new__*(), *__reduce__*(), *__reduce_ex__*(), *__repr__*(), *__setattr__*(), *__sizeof__*(), *__str__*(), *__subclasshook__*()

93.7.2 Properties

Name	Description
<i>Inherited from object</i>	
<i>__class__</i>	

93.7.3 Class Variables

Name	Description
<code>__doc__</code>	Value: <code>"""This class generates a map from inputs to intege...</code>

94 Module `gmisclib.nicknames`

94.1 Functions

nicknames(*namelist*)

Takes a list of names and trims off junk from the beginning and ends of the names to produce a set of reasonable nicknames that are relatively compact and easy to read.

It returns a map from the names to the nicknames, along with the stuff trimmed off from the edges.

test()

94.2 Variables

Name	Description
<code>--package--</code>	Value: None

95 Module *gmisclib.nmf*

$V=WH$, where W and H are non-negative.

From D. D. Lee and H. S. Seung, 'Learning the parts of objects by nonnegative matrix factorization.'

95.1 Functions

nmf (<i>v</i> , <i>rank</i>)

95.2 Variables

Name	Description
EPS	Value: 1e-06
CC	Value: 2
IEPS	Value: 0.03
__package__	Value: 'gmisclib'

96 Module *gmisclib.opt*

This module is a Levenberg-Marquardt optimizer with stem-size control for numeric differentiation. It just needs a function that produces residuals. It is multi-threaded, so calculations of residuals can be farmed out to many processors.

UPDATED 12/2009 GPK: NOT TESTED!

96.1 Functions

vec_equal(*a*, *b*)

Check that two Numeric vectors are exactly equal.

sumsq(*a*)

This is the overall error measure.

maxabs(*a*)

This is used as a measure of how much the function changed as a result of a parameter change.

wtf(*r_dz*, *delta*, *quantum*)

This shows how important different measurements are to the final derivative estimate. The weight must be small when the change of a parameter is smaller than the quantum (i.e. when $r_delta < quantum$). It must also be small when the change in z is much larger than T ($r_dz >> 1$). It is largest when r_dz is about 1.

near_duplicate(*guess*, *tmp*, *quantum*)

symmetry_point(*tmp*, *T*, *quantum*)

Given a list of ($delta$, z) in a differentiation, pick one more point that makes the differentiation more symmetric. This is done by picking a new $delta$ that brings $\sum\{delta_i * wtf()\}$ closer to zero, where $wtf()$ is the weight given to each point in the differentiation.

scale_est(*tmp*, *T*)

explored_region(*tmp*)

clz_point(*qq*, *T*, *quantum*)

Given a list of (delta, z) tuples in a differentiation attempt, find another delta that (a) fills a gap in the sequence, (b) is preferably has a sign opposite most of the {delta} values, and (c) has a large wtf(). Points outside the sequence are also considered.

deriv_estimate(*tmp*, *T*, *quantum*)

need_more_diff_pts(*tmp*, *T*, *quantum*)

diff_one(*x*, *i*, *sem*)

Differentiate z with respect to parameter i. We enter this function with the semaphore 'sem' already acquired.

eval_lambda(*processor*, *a*, *b*, *startp*, *lamb*, *out*, *opt*)

lamb_correct(*newscale*, *scale*)

anneal_guts(*p*, *rv*, *T*, *lamb*, *sem*, *opt*)

test1_fcn(*p*, *args*, **d*)

test1()

Linear.

test1a_fcn(*p*, *args*, **d*)

test1a()

Linear.

test2_fcn(*p*, *args*, **d*)

test2()

Nonlinear

test3_fcn(*p*, *args*, **d*)

test3()

Linear, but correlated

test4_fcn(*p*, *args*, **d*)**test4()**

Linear, but with a constraint

test5_fcn(*p*, *args*, **d*)**test5()**

Very Nonlinear

test6_fcn(*p*, *args*, **d*)**test6()**

Curved valley.

linconst_min(*nparam*, *param*, *min*)

Generate a linear constraint to be added onto a list and passed to `linear_constraint()`. This constraint expresses that `p[param]>=min`.

linconst_max(*nparam*, *param*, *max*)

Generate a linear constraint to be added onto a list and passed to `linear_constraint()`. This constraint expresses that `p[param]<=max`.

linear_constraint(*op*, *np*, *list_of_constraints*)

This constrains a step to lie inside a region bounded by a `list_of_constraints`.

Parameters

op: parameters at the beginning of the step.

np: target parameters.

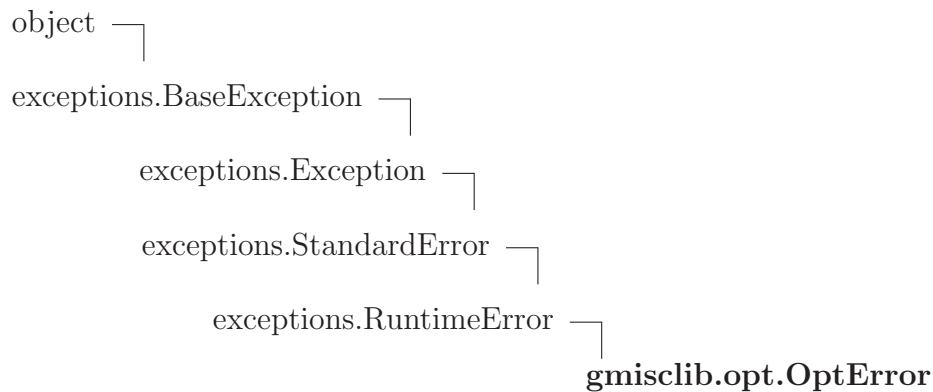
list_of_constraints: [(*vector*, *shift*), ...] where the allowable region of each constraint is defined by `Num.dot(x, vector)+shift >= 0`. *Vector* is a numpy array.

test7()
Constraint.

96.2 Variables

Name	Description
<code>--package--</code>	Value: <code>'gmislib'</code>

96.3 Class *OptError*



Known Subclasses: *gmislib.opt.BadResult*, *gmislib.opt.NoDerivative*, *gmislib.opt.NoDownhill*

96.3.1 Methods

<code>--init--(self, s=None)</code>
<code>x.__init__(...)</code> initializes <code>x</code> ; see <code>x.__class__.__doc__</code> for signature
Overrides: <code>object.__init__</code> extit(inherited documentation)

Inherited from exceptions.RuntimeError

`--new--()`

Inherited from exceptions.BaseException

`--delattr--()`, `--getattribute--()`, `--getitem--()`, `--getslice--()`, `--reduce--()`, `--repr--()`,
`--setattr--()`, `--setstate--()`, `--str--()`, `--unicode--()`

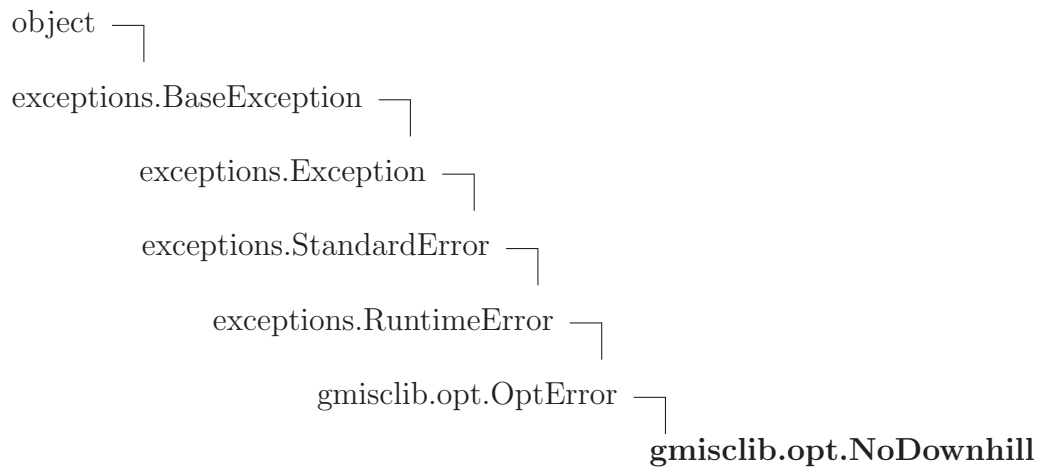
Inherited from object

`--format--()`, `--hash--()`, `--reduce_ex--()`, `--sizeof--()`, `--subclasshook--()`

96.3.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
args, message	
<i>Inherited from object</i>	
__class__	

96.4 Class NoDownhill



96.4.1 Methods

```

__init__(self, s=None)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature
Overrides: object.__init__ extit(inherited documentation)

```

Inherited from exceptions.RuntimeError

__new__()

Inherited from exceptions.BaseException

__delattr__(), __getattr__(), __getitem__(), __getslice__(), __reduce__(), __repr__(),
 __setattr__(), __setstate__(), __str__(), __unicode__()

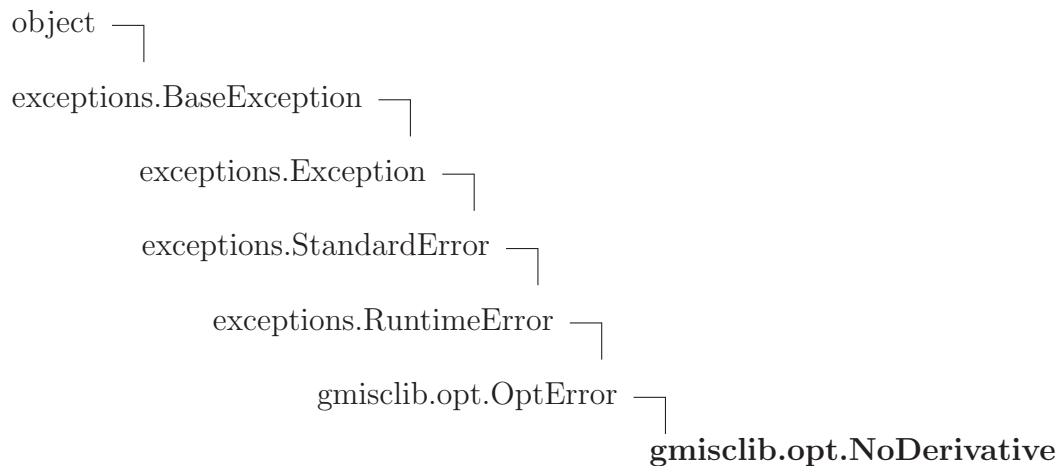
Inherited from object

__format__(), __hash__(), __reduce_ex__(), __sizeof__(), __subclasshook__()

96.4.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
args, message	
<i>Inherited from object</i>	
__class__	

96.5 Class NoDerivative



96.5.1 Methods

```

__init__(self, s=None)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature
Overrides: object.__init__ extit(inherited documentation)

```

Inherited from exceptions.RuntimeError

__new__()

Inherited from exceptions.BaseException

__delattr__(), __getattr__(), __getitem__(), __getslice__(), __reduce__(), __repr__(),
 __setattr__(), __setstate__(), __str__(), __unicode__()

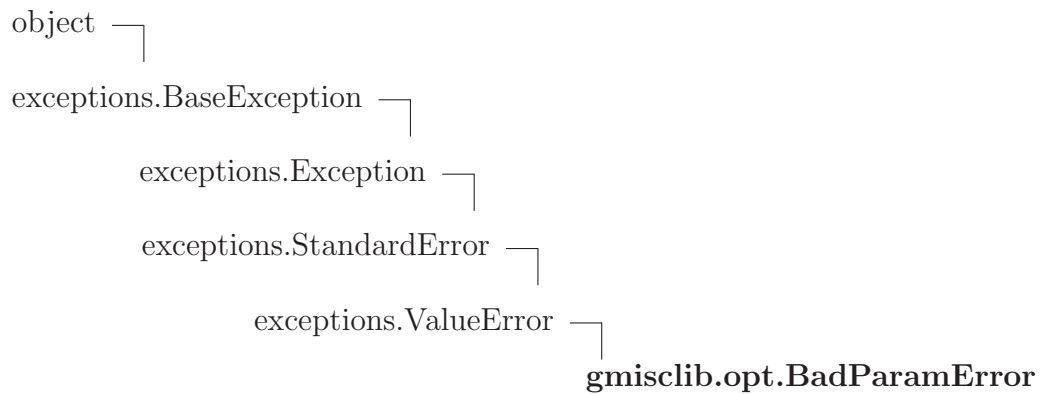
Inherited from object

__format__(), __hash__(), __reduce_ex__(), __sizeof__(), __subclasshook__()

96.5.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
<code>args</code> , <code>message</code>	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

96.6 Class `BadParamError`



Raised when you give and opt instance some invalid control parameter.

96.6.1 Methods

```

__init__(self, s=None)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature
Overrides: object.__init__ extit(inherited documentation)

```

Inherited from `exceptions.ValueError`

```
__new__()
```

Inherited from `exceptions.BaseException`

```

__delattr__(), __getattr__(), __getitem__(), __getslice__(), __reduce__(), __repr__(),
__setattr__(), __setstate__(), __str__(), __unicode__()

```

Inherited from `object`

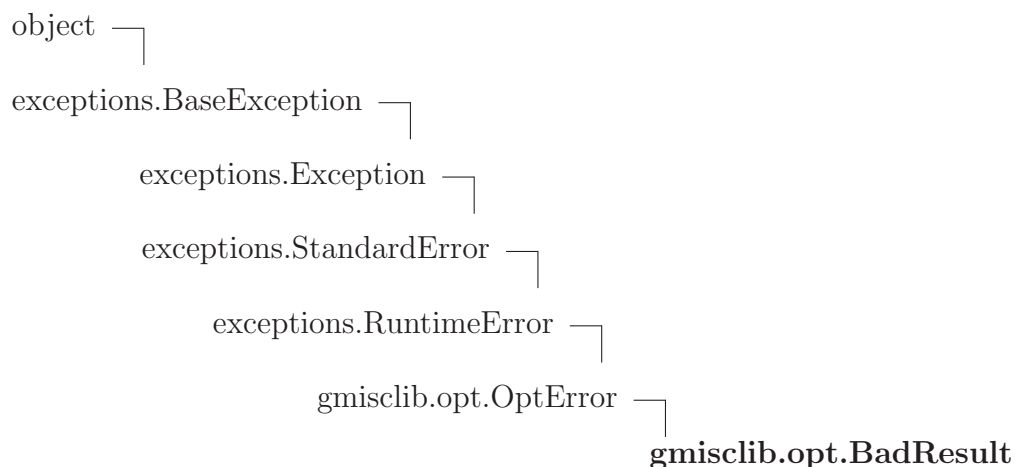
```
__format__(), __hash__(), __reduce_ex__(), __sizeof__(), __subclasshook__()
```

96.6.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
args, message	
<i>Inherited from object</i>	
__class__	

96.6.3 Class Variables

Name	Description
__doc__	Value: <code>""</code> "Raised when you give and opt instance some invalid con..."

96.7 Class *BadResult*

The function to be optimized returns some illegal result. For instance, an array of the wrong size or type, or it returns None for the initial evaluation.

96.7.1 Methods

```

__init__(self, s=None)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature
Overrides: object.__init__ extit(inherited documentation)

```

Inherited from exceptions.RuntimeError

`__new__()`

Inherited from exceptions.BaseException

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

Inherited from object

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

96.7.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
args, message	
<i>Inherited from object</i>	
__class__	

96.7.3 Class Variables

Name	Description
__doc__	Value: <code>""</code> "The function to be optimized returns some illegal resu..."

96.8 Class mysem

This is a semaphore that is automatically released when it is de-allocated. It also contains a processor ID.

96.8.1 Methods

`__init__(self, id, sem)`

`release(self)`

`__del__(self)`

96.8.2 Class Variables

Name	Description
<code>--doc--</code>	Value: <code>""</code> "This is a semaphore that is automatically released w..."

96.9 Class semclass

This is a semaphore to control the number of simultaneous computations. It hands out a mysem class which is used to release the semaphore.

96.9.1 Methods

<code>--init--(self, n)</code>
<code>acquire(self)</code>

96.9.2 Class Variables

Name	Description
<code>--doc--</code>	Value: <code>""</code> "This is a semaphore to control the number..."

96.10 Class prms

This class records parameters and caches function evaluations. All function evaluations happen here.

96.10.1 Methods

<code>--init--(self, p, fn, args, processor, diffparam=None, diffctr=None)</code>
<code>zraw(self)</code>
<code>z(self)</code>
<code>sumsq(self)</code>

`__len__(self)``__str__(self)``__repr__(self)`

96.10.2 Class Variables

Name	Description
<code>__doc__</code>	Value: <code>"""This class records parameters and caches function eval...</code>

96.11 Class *LockedList*

96.11.1 Methods

`__init__(self, start=None)``append(self, x)``__len__(self)``pop(self, i=-1)``items(self)`

96.12 Class *opt*

A class that implements a optimizer.

96.12.1 Methods

__init__(*self*, *p*, *T*, *fcn*, *args*=None)

Create an optimizer object.

You can also set the following variables between calls to `self.step()`:

`C{scale}`, `C{deriv_quantum}`, `C{T}`, `C{active}`, `C{sem}`, `C{constrain}`

@param *p*: initial parameter vector.

@param *T*: temperature for simulated annealing and numeric differentiation.

@param *fcn*: `fcn(p, args, diffparam, diffctr, processor)`, where

p = parameters, *args* = arbitrary reference passed to function,
 diffparam = None (unless differentiating),
 diffctr = None (unless differentiating),
 processor = (*proc_id*, *i_steps*),

where *proc_id* is in `range(self.nthreads)`
 and specifies which processor ought
 to be assigned to the task, and
 i_steps is the number of steps the
 optimization has gone through.

The function is assumed to return either a Num Python array
 or a sequence of arrays and floats, mixed.

@ivar *scale*: estimate of the step size to use in differentiation.

@ivar *deriv_quantum*: minimum step size used in differentiation.

@ivar *T*: simulated annealing temperature.

@ivar *active*: vector of which parameters to optimize. Only the active
 parameters change their value.

@ivar *sem*: a semaphore to control the number of threads to use
 (if you want to set the # of threads do `self.sem = semclass(nnn)`).

@ivar *constrain*: a function that constrains the step.

You can use it to do a constrained fit by having it project
 the proposed step back into the legal volume.

Called as `self.constrain(old_prms, proposed_prms, self.args)`.

It returns the constrained step, or None if there isn't any.

set_nthreads(*self*, *n*)

Set the number of simultaneous threads to be allowed. Calculations of the
 residuals are farmed out to threads.

z(*self*)

What is the current residual?

sumsq(*self*)

What is the current error?

print_ev_diag(*self*, *misc*={}, *vectors*=0)**Td**(*self*)

This allows the differentiation temperature to be dynamically set.

dE(*self*)**quick_lambda**(*self*, *a*, *b*, *startp*, *lamb*, *ntries*=10)**search_with_update**(*self*, *lamb*, *newp*)**calc_curv**(*self*)**measure**(*self*)

If you decrease the number of active parameters, it might behoove you to set `self.diff=None`, before you call this function, so that old derivatives get flushed.

step(*self*)

A minimization step.

predicted_sumsq_delta(*self*, *last_p*)

The energy change of a step, assuming you're at the minimum.

one_anneal(*self*, *T*)

Take one simulated annealing step, as fast as possible. We send all the processors in random directions, and the one that reports an acceptable step first is returned. Returns 1 on success, 0 if no good step could be found in a reasonable number of tries. The temperature can also be a function of one argument (as an alternative to a float), in which case, the temperature is calculated as `T(self)`.

terminate(*self*)

Returns number between 0 and 1 if the optimization seems finished. Returns -1 if it's clearly not done yet. This is designed to be called after `step()`, so that both `self.p` and `self.last_p` refer to the results of downhill steps.

covariance_under(*self*)

This is an underestimate of covariance. Adding in the `self.lamb` term means that it attempts to correct for the nonlinearity of the problem, at least along the direction of search.

covariance(*self*)

Covariance estimate.

run(*self*, *maxsteps*=None, *misc*={}, *Ta*=None)

This is the normal top-level routine. You should feel free to make your own, though.

print_errbar(*self*)

sim_anneal(*self*, *nsteps*, *Ta*=None, *misc*={})

A top-level simulated annealing routine.

print_at(*self*, *misc*={})

Prints a file to match the old C++ optimizer output.

96.12.2 Class Variables

Name	Description
<code>__doc__</code>	Value: <code>"""A class that implements a optimizer. """</code>
<code>LOG_NAME</code>	Value: <code>'a.tmp'</code>

97 Module *gmisclib.ortho_poly*

97.1 Functions

test (<i>name</i>)

F (<i>name</i> , <i>n</i> =None, <i>x</i> =None)
--

This is a factory function. Get any kind of ortho poly you want, as selected by the first argument.

97.2 Variables

Name	Description
<code>--package--</code>	Value: 'gmisclib'

97.3 Class *ortho*

Known Subclasses: *gmisclib.ortho_poly.ortho_poly*, *gmisclib.ortho_poly.SLTB*, *gmisclib.ortho_poly.SinCo*

97.3.1 Methods

__init__ (<i>self</i> , <i>n</i> =None, <i>x</i> =None)

Argument <i>n</i> is how many points you want the polynomials evaluated at. The ordinates of the points are placed in <i>self.x</i> . You may alternatively specify the points as argument <i>x</i> , in which case the <i>x_()</i> function will never be called.
--

wt (<i>self</i>)

Weighting function to get orthonormality. This sets <i>self._numwt</i> .
--

x_ (<i>self</i> , <i>m</i>)

Calculates the points at which the function is evaluated, if you want <i>m</i> evenly spaced points. By default, the function is assumed to range over the open interval (-1, 1). You can override this function in a subclass to get a different range.
--

P(*self*, *i*)

Self.P(*i*) is the *i*th orthogonal polynomial. The result is normalized so that `numpy.sum(self.wt() * self.P(i)**2) == 1`. It returns an un-shared array, so the array can be modified without affecting future calls to P(*i*).

expand(*self*, *c*)**compute**(*self*, *n*)**wt**_*(self)*

97.3.2 Class Variables

Name	Description
registry	Value: {}

97.4 Class `ortho_poly`

gmisclib.ortho_poly.ortho —
gmisclib.ortho_poly.ortho_poly

Known Subclasses: gmisclib.ortho_poly.Chebyshev, gmisclib.ortho_poly.Chebyshev2, gmisclib.ortho_poly.

Virtual base class. This is a superclass of all orthorgonal polynomials. This does the recurrence [Abramowitz and Stegun p.782], and ensures that the generated polynomials are all orthonormal to each other. The derived classed need to specify two functions: `recurse()` and `wt_()`. `Recurse()` is the recursion relation from P(*i*) and P(*i*-1) to P(*i*+1), and `wt_()` is the weighting function for the polynomial. `Wt_()` is needed to check orthogonality, and really defines the polynomial. These polynomials are guarenteed to be orthogonal to eachother when summed over the supplied set of *x* points. Thus, if you change `x_()` or `wt_()`, you get a different set of functions.

97.4.1 Methods

`__init__(self, n=None, x=None)`

Argument *n* is how many points you want the polynomials evaluated at. The ordinates of the points are placed in *self.x*. You may alternatively specify the points as argument *x*, in which case the *x_()* function will never be called.

Overrides: *gmisclib.ortho_poly.ortho.__init__* *exitit*(inherited documentation)

`P(self, i)`

Self.P(i) is the *i*th orthogonal polynomial. The result is normalized so that `numpy.sum(self.P(i)**2) == 1`.

Overrides: *gmisclib.ortho_poly.ortho.P*

`recurse(self, n, fn, fnm1)`

Does the recurrence relation, evaluating *f_n(n+1)* as a function of *n*, *f_n*, and *f_{n-1}*. For *n=0* and *n=1*, *fn* and *fnm1* may be *None*.

`compute(self, n)`

Overrides: *gmisclib.ortho_poly.ortho.compute*

`expand(self, c)`

`wt(self)`

Weighting function to get orthonormality. This sets *self._numwt*.

`wt_(self)`

`x_(self, m)`

Calculates the points at which the function is evaluated, if you want *m* evenly spaced points. By default, the function is assumed to range over the open interval *(-1, 1)*. You can override this function in a subclass to get a different range.

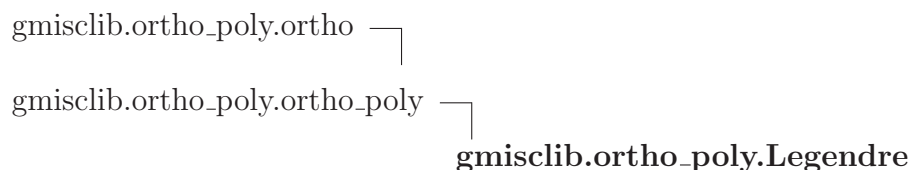
97.4.2 Class Variables

Name	Description
<code>__doc__</code>	Value: <code>"""Virtual base cla...</code>

continued on next page

Name	Description
registry	Value: {}

97.5 Class Legendre



Legendre polynomials, orthonormal over $(-1, 1)$ with weight 1. Called $P_n(x)$ in Abramowitz and Stegun.

97.5.1 Methods

`__init__(self, n=None, x=None)`

Argument `n` is how many points you want the polynomials evaluated at. The ordinates of the points are placed in `self.x`. You may alternatively specify the points as argument `x`, in which case the `x_()` function will never be called.

Overrides: `gmisclib.ortho_poly.ortho.__init__` `exitit`(inherited documentation)

`recurse(self, n, fn, fnm1)`

Does the recurrence relation, evaluating $f_{(n+1)}$ as a function of n , f_n , and $f_{(n-1)}$. For $n=0$ and $n=1$, `fn` and `fnm1` may be `None`.

Overrides: `gmisclib.ortho_poly.ortho_poly.recurse` `exitit`(inherited documentation)

`wt_(self)`

Overrides: `gmisclib.ortho_poly.ortho.wt_`

`P(self, i)`

`Self.P(i)` is the i th orthogonal polynomial. The result is normalized so that `numpy.sum(self.P(i)**2) == 1`.

Overrides: `gmisclib.ortho_poly.ortho.P`

compute(*self*, *n*)

Overrides: gmisclib.ortho_poly.ortho.compute

expand(*self*, *c*)

wt(*self*)

Weighting function to get orthonormality. This sets self._numwt.

x_(*self*, *m*)

Calculates the points at which the function is evaluated, if you want *m* evenly spaced points. By default, the function is assumed to range over the open interval (-1, 1). You can override this function in a subclass to get a different range.

97.5.2 Class Variables

Name	Description
<code>__doc__</code>	Value: <code>""Legendre polynomia...</code>
<code>name</code>	Value: <code>'Legendre'</code>
<code>registry</code>	Value: <code>{}</code>

97.6 Class Chebyshev

gmisclib.ortho_poly.ortho

gmisclib.ortho_poly.ortho_poly

gmisclib.ortho_poly.Chebyshev

Chebyshev polynomials of the first kind, orthonormal over (-1, 1) with weight $(1-x^2)^{-1/2}$. These are the equi-ripple polynomials. Called $T_n(x)$ in Abramowitz and Stegun.

97.6.1 Methods

`__init__(self, n=None, x=None)`

Argument `n` is how many points you want the polynomials evaluated at. The ordinates of the points are placed in `self.x`. You may alternatively specify the points as argument `x`, in which case the `x_()` function will never be called.

Overrides: `gmisclib.ortho_poly.ortho.__init__` `exitit`(inherited documentation)

`recurse(self, n, fn, fnm1)`

Does the recurrence relation, evaluating `f_(n+1)` as a function of `n`, `f_n`, and `f_(n-1)`. For `n=0` and `n=1`, `fn` and `fnm1` may be `None`.

Overrides: `gmisclib.ortho_poly.ortho_poly.recurse` `exitit`(inherited documentation)

`wt_(self)`

Overrides: `gmisclib.ortho_poly.ortho.wt_`

`P(self, i)`

`Self.P(i)` is the `i`th orthogonal polynomial. The result is normalized so that `numpy.sum(self.P(i)**2) == 1`.

Overrides: `gmisclib.ortho_poly.ortho.P`

`compute(self, n)`

Overrides: `gmisclib.ortho_poly.ortho.compute`

`expand(self, c)`

`wt(self)`

Weighting function to get orthonormality. This sets `self._numwt`.

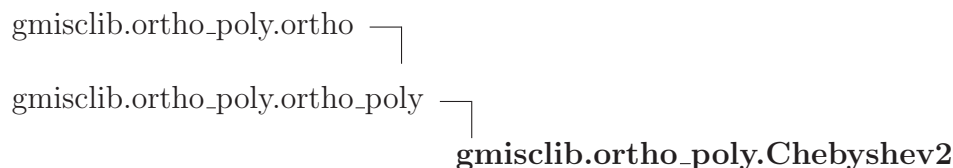
`x_(self, m)`

Calculates the points at which the function is evaluated, if you want `m` evenly spaced points. By default, the function is assumed to range over the open interval `(-1, 1)`. You can override this function in a subclass to get a different range.

97.6.2 Class Variables

Name	Description
<code>__doc__</code>	Value: <code>"""Chebyshev polynomials of the first ki...</code>
<code>name</code>	Value: <code>'Chebyshev'</code>
<code>registry</code>	Value: <code>{}</code>

97.7 Class *Chebyshev2*



Chebyshev polynomials of the second kind, orthonormal over $(-1, 1)$ with weight $(1-x^2)^{1/2}$. Called $Un(x)$ in Abramowitz and Stegun.

97.7.1 Methods

`__init__(self, n=None, x=None)`

Argument `n` is how many points you want the polynomials evaluated at. The ordinates of the points are placed in `self.x`. You may alternatively specify the points as argument `x`, in which case the `x_()` function will never be called.

Overrides: `gmisclib.ortho_poly.ortho.__init__` `exitit`(inherited documentation)

`recurse(self, n, fn, fnm1)`

Does the recurrence relation, evaluating f_{n+1} as a function of n , f_n , and f_{n-1} . For $n=0$ and $n=1$, `fn` and `fnm1` may be `None`.

Overrides: `gmisclib.ortho_poly.ortho_poly.recurse` `exitit`(inherited documentation)

`wt_(self)`

Overrides: `gmisclib.ortho_poly.ortho.wt_`

P(*self*, *i*)

Self.P(i) is the ith orthogonal polynomial. The result is normalized so that `numpy.sum(self.P(i)**2) == 1`.

Overrides: `gmisclib.ortho_poly.ortho.P`

compute(*self*, *n*)

Overrides: `gmisclib.ortho_poly.ortho.compute`

expand(*self*, *c*)

wt(*self*)

Weighting function to get orthonormality. This sets `self._numwt`.

x_(*self*, *m*)

Calculates the points at which the function is evaluated, if you want *m* evenly spaced points. By default, the function is assumed to range over the open interval $(-1, 1)$. You can override this function in a subclass to get a different range.

97.7.2 Class Variables

Name	Description
<code>__doc__</code>	Value: <code>""</code> "Chebyshev polynomials of the second ki...
<code>name</code>	Value: <code>'Chebyshev2'</code>
<code>registry</code>	Value: <code>{}</code>

97.8 Class SinCos

`gmisclib.ortho_poly.ortho` └

`gmisclib.ortho_poly.SinCos`

1, $\sin(\pi x)$, $\cos(\pi x)$, $\sin(2\pi x)$, $\cos(2\pi x)$... orthonormal over $(-1, 1)$ with weight 1.

97.8.1 Methods

__init__(*self*, *n*=None, *x*=None)

Argument *n* is how many points you want the polynomials evaluated at. The ordinates of the points are placed in *self.x*. You may alternatively specify the points as argument *x*, in which case the *x_()* function will never be called.

Overrides: gmisclib.ortho_poly.ortho.__init__ *exitit*(inherited documentation)

compute(*self*, *n*)

Overrides: gmisclib.ortho_poly.ortho.compute

wt_(*self*)

Overrides: gmisclib.ortho_poly.ortho.wt_

P(*self*, *i*)

Self.P(*i*) is the *i*th orthogonal polynomial. The result is normalized so that `numpy.sum(self.wt() * self.P(i)**2) == 1`. It returns an un-shared array, so the array can be modified without affecting future calls to P(*i*).

expand(*self*, *c*)

wt(*self*)

Weighting function to get orthonormality. This sets *self.numwt*.

x_(*self*, *m*)

Calculates the points at which the function is evaluated, if you want *m* evenly spaced points. By default, the function is assumed to range over the open interval (-1, 1). You can override this function in a subclass to get a different range.

97.8.2 Class Variables

Name	Description
<code>__doc__</code>	Value: <code>"""1, sin(pi*x), cos(pi*x), sin(2*pi*x), cos(2*pi*x)...</code>
<code>name</code>	Value: <code>'SinCos'</code>
<code>registry</code>	Value: <code>{}</code>

97.9 Class SLTB

gmsclib.ortho_poly.ortho —
gmsclib.ortho_poly.SLTB

Smooth Local Trigonometric Basis. From Björn Jawerth, Yi Liu, Wim Sweldens, "Signal Compression with Smooth Local Basis Functions".

97.9.1 Methods

__init__(*self*, *n*=None, *x*=None, *eps*=0.5)

Argument *n* is how many points you want the polynomials evaluated at. The ordinates of the points are placed in *self.x*. You may alternatively specify the points as argument *x*, in which case the *x_()* function will never be called.

Overrides: gmsclib.ortho_poly.ortho.__init__ extit(inherited documentation)

compute(*self*, *n*)

Overrides: gmsclib.ortho_poly.ortho.compute

r(*self*, *x*)

This must have $l^{*2}+r^{*2}=1$

b(*self*, *x*)

wt_(*self*)

Overrides: gmsclib.ortho_poly.ortho.wt_

x_(*self*, *m*)

Calculates the points at which the function is evaluated, if you want *m* evenly spaced points. The function is assumed to range over the open interval $(-eps, 1+eps)$. You can override this function in a subclass to get a different range.

Overrides: gmsclib.ortho_poly.ortho.x_

test()

P(*self*, *i*)

Self.P(i) is the ith orthogonal polynomial. The result is normalized so that `numpy.sum(self.wt() * self.P(i)**2) == 1`. It returns an un-shared array, so the array can be modified without affecting future calls to P(i).

expand(*self*, *c*)

wt(*self*)

Weighting function to get orthonormality. This sets self._numwt.

97.9.2 Class Variables

Name	Description
<code>--doc--</code>	Value: <code>""Smooth Local Trigonometric Bas...</code>
<code>name</code>	Value: <code>'SLTB'</code>
<code>registry</code>	Value: <code>{}</code>

98 Module `gmisclib.parse_tree_number`

98.1 Functions

`is_mul_of(a, b, indent='')`

Returns the ratio `a/b` if `a` contains `b` as a factor, returns `None` otherwise. The routine does not promise to find all possible factors.

`abs(x)`

`cos(x)`

`sin(x)`

`exp(x)`

`log(x)`

`sqrt(x)`

`coerce_into(x)`

`operatorN(operator, operands)`

98.2 Variables

Name	Description
<code>__package__</code>	Value: <code>'gmisclib'</code>
<code>__warningregistry__</code>	Value: <code>{('the sets module is deprecated', <type 'exceptions.Depr...</code>

98.3 Class Array

object —
 `gmisclib.parse_tree_number.Array`

98.3.1 Methods

<code>__init__</code> (<i>self</i> , <i>name</i>)
--

This creates an array. Arrays are basic elements of expressions.
--

Overrides: <code>object.__init__</code>

<code>__getitem__</code> (<i>self</i> , <i>key</i>)
--

<code>__str__</code> (<i>self</i> , <i>env</i> ={})

<code>str(x)</code>

Overrides: <code>object.__str__</code> <code>exitit</code> (inherited documentation)
--

<code>__repr__</code> (<i>self</i> , <i>env</i> ={})
--

<code>str(x)</code>

Overrides: <code>object.__repr__</code> <code>exitit</code> (inherited documentation)

<code>indices</code> (<i>self</i> , <i>env</i> ={})

<code>eval</code> (<i>self</i> , <i>env</i> ={})
--

<code>variables</code> (<i>self</i> , <i>env</i> ={})

<code>debug</code> (<i>self</i>)

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__setattr__()`, `__sizeof__()`, `__subclasshook__()`

98.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

98.4 Class `output_mixin`

object —
 `gmislib.parse_tree_number.output_mixin`

Known Subclasses: `gmislib.parse_tree_number.Expression`, `gmislib.parse_tree_number.Float`, `gmislib.parse_tree_number.Name`, `gmislib.parse_tree_number._operator`

98.4.1 Methods

`__init__(self)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` `exitit`(inherited documentation)

`eval(self, env={})`

This returns an `Expression`.

`variables(self, env={})`

Any definitions in `env` are used to resolve otherwise undefined names. This returns a `Set` containing all undefined variable names.

`indices(self, variable, env={})`

`debug(self)`

`__str__(self, env={})`

`Env` is optional. Anything defined in `env` is used to resolve references, but the outcome may still contain undefined variables, which are printed as names.

Overrides: `object.__str__`

Inherited from `object`

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__repr__()`, `__setattr__()`, `__sizeof__()`, `__subclasshook__()`

98.4.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

98.5 Class `abstract_number`

object └─ `gmisclib.parse_tree_number.abstract_number`

Known Subclasses: `gmisclib.parse_tree_number.Expression`, `gmisclib.parse_tree_number.Float`, `gmisclib.parse_tree_number.Name`, `gmisclib.parse_tree_number._operator`

98.5.1 Methods

`__init__(self)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` extit(inherited documentation)

`__add__(self, other)`

`__radd__(self, other)`

`__sub__(self, other)`

`__rsub__(self, other)`

`__pow__(self, other)`

`__rpow__(self, other)`

`__mul__(self, other)`

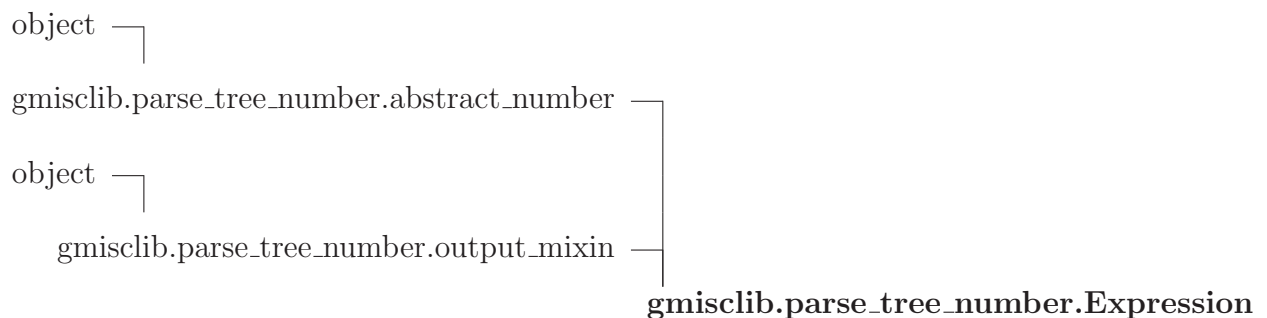
`__rmul__(self, other)`

`__div__(self, other)`

`__rdiv__(self, other)`

`--truediv--(self, other)``--rtruediv--(self, other)``--neg--(self)``--pos--(self)``--abs--(self)`***Inherited from object***`--delattr--()`, `--format--()`, `--getattr--()`, `--hash--()`, `--new--()`, `--reduce--()`, `--reduce_ex--()`,
`--repr--()`, `--setattr--()`, `--sizeof--()`, `--str--()`, `--subclasshook--()`**98.5.2 Properties**

Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

98.6 Class Expression**98.6.1 Methods**`--init--(self, value)``x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signatureOverrides: `object.__init__` `extit`(inherited documentation)

```
__iadd__(self, other)
```

```
__isub__(self, other)
```

```
__imul__(self, other)
```

```
__idiv__(self, other)
```

```
eval(self, env={})
```

This returns an Expression.

Overrides: gmisclib.parse_tree_number.output_mixin.eval extit(inherited documentation)

```
variables(self, env={})
```

Any definitions in env are used to resolve otherwise undefined names. This returns a Set containing all undefined variable names.

Overrides: gmisclib.parse_tree_number.output_mixin.variables extit(inherited documentation)

```
indices(self, variable, env={})
```

Overrides: gmisclib.parse_tree_number.output_mixin.indices

```
__str__(self, env={})
```

Env is optional. Anything defined in env is used to resolve references, but the outcome may still contain undefined variables, which are printed as names.

Overrides: object.__str__ extit(inherited documentation)

```
__repr__(self, env={})
```

Env is optional. Anything defined in env is used to resolve references, but the outcome may still contain undefined variables, which are printed as names.

Overrides: object.__repr__ extit(inherited documentation)

```
debug(self)
```

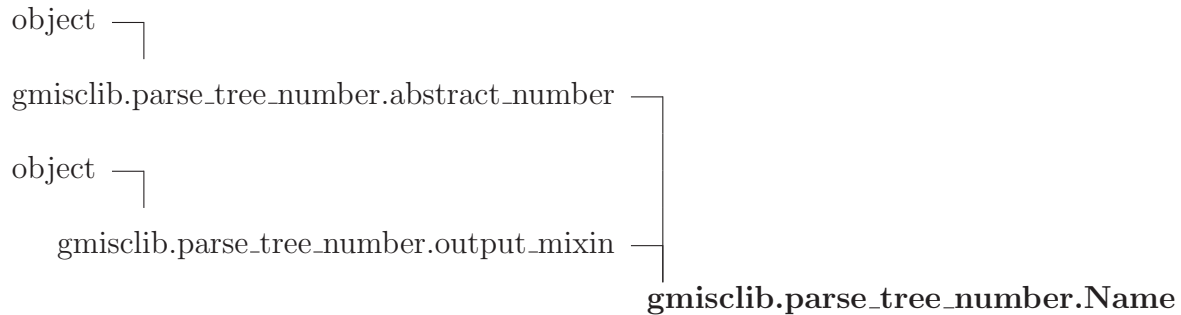
Overrides: gmisclib.parse_tree_number.output_mixin.debug

```
__float__(self)
```

`priority(self)``__abs__(self)``__add__(self, other)``__div__(self, other)``__mul__(self, other)``__neg__(self)``__pos__(self)``__pow__(self, other)``__radd__(self, other)``__rdiv__(self, other)``__rmul__(self, other)``__rpow__(self, other)``__rsub__(self, other)``__rtruediv__(self, other)``__sub__(self, other)``__truediv__(self, other)`***Inherited from object***`__delattr__(), __format__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
__setattr__(), __sizeof__(), __subclasshook__()`**98.6.2 Properties**

Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

98.7 Class Name



98.7.1 Methods

<code>--init--</code> (<i>self</i> , <i>name</i>)
This creates a named variable. Names are basic parts of expressions. Overrides: <code>object.__init__</code>
<code>--str--</code> (<i>self</i> , <i>env</i> ={})
Env is optional. Anything defined in <i>env</i> is used to resolve references, but the outcome may still contain undefined variables, which are printed as names. Overrides: <code>object.__str__</code> <code>exitit</code> (inherited documentation)
<code>--repr--</code> (<i>self</i> , <i>env</i> ={})
Env is optional. Anything defined in <i>env</i> is used to resolve references, but the outcome may still contain undefined variables, which are printed as names. Overrides: <code>object.__repr__</code> <code>exitit</code> (inherited documentation)
<code>eval</code> (<i>self</i> , <i>env</i>)
This returns an Expression. Overrides: <code>gmisclib.parse_tree_number.output_mixin.eval</code> <code>exitit</code> (inherited documentation)

variables(*self*, *env*={})

Any definitions in *env* are used to resolve otherwise undefined names. This returns a Set containing all undefined variable names.

Overrides: `gmisclib.parse_tree_number.output_mixin.variables` `exitit`(inherited documentation)

indices(*self*, *variable*, *env*={})

Overrides: `gmisclib.parse_tree_number.output_mixin.indices`

debug(*self*)

Overrides: `gmisclib.parse_tree_number.output_mixin.debug`

__abs__(*self*)**__add__**(*self*, *other*)**__div__**(*self*, *other*)**__mul__**(*self*, *other*)**__neg__**(*self*)**__pos__**(*self*)**__pow__**(*self*, *other*)**__radd__**(*self*, *other*)**__rdiv__**(*self*, *other*)**__rmul__**(*self*, *other*)**__rpow__**(*self*, *other*)**__rsub__**(*self*, *other*)**__rtruediv__**(*self*, *other*)


```
--sub--(self, other)
```

```
--truediv--(self, other)
```

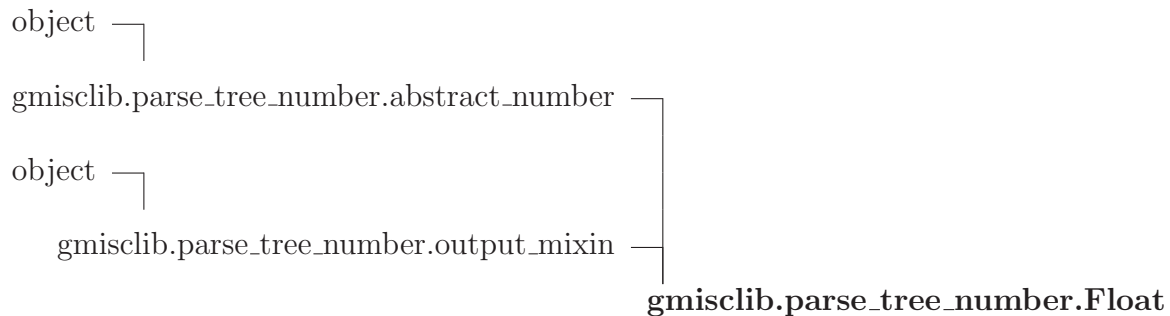
Inherited from object

```
--delattr--(), --format--(), --getattr--(), --hash--(), --new--(), --reduce--(), --reduce_ex--(),
--setattr--(), --sizeof--(), --subclasshook--()
```

98.7.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

98.8 Class Float



98.8.1 Methods

```
--init--(self, v)
```

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` extit(inherited documentation)

```
eval(self, env={})
```

This returns an Expression.

Overrides: `gmisclib.parse_tree_number.output_mixin.eval` extit(inherited documentation)

variables(*self*, *env*={})

Any definitions in *env* are used to resolve otherwise undefined names. This returns a Set containing all undefined variable names.

Overrides: *gmslib.parse_tree.number.output_mixin.variables* *exitit*(inherited documentation)

indices(*self*, *variable*, *env*={})

Overrides: *gmslib.parse_tree.number.output_mixin.indices*

debug(*self*, *env*={})

Overrides: *gmslib.parse_tree.number.output_mixin.debug*

__str__(*self*, *env*={})

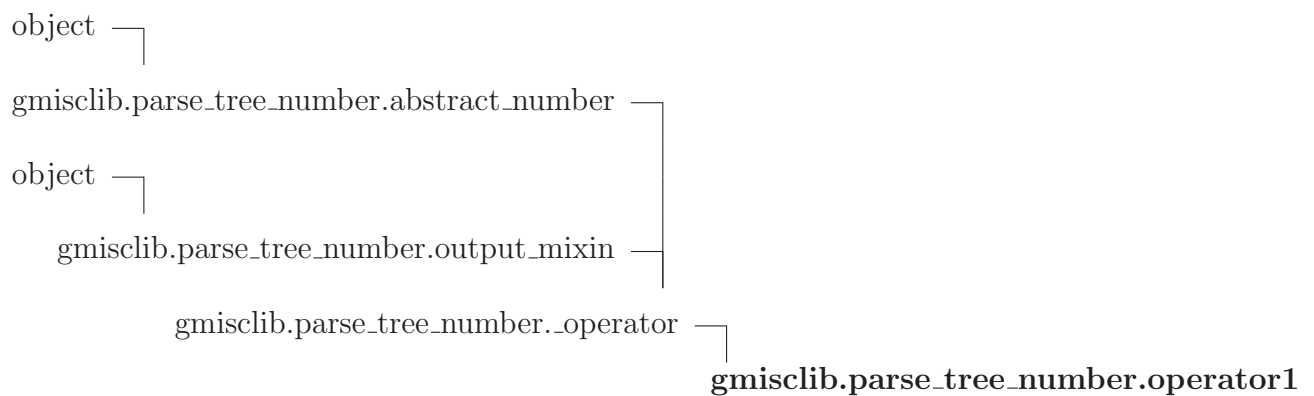
Env is optional. Anything defined in *env* is used to resolve references, but the outcome may still contain undefined variables, which are printed as names.

Overrides: *object.__str__* *exitit*(inherited documentation)

__float__(*self*)**__abs__**(*self*)**__add__**(*self*, *other*)**__div__**(*self*, *other*)**__mul__**(*self*, *other*)**__neg__**(*self*)**__pos__**(*self*)**__pow__**(*self*, *other*)**__radd__**(*self*, *other*)**__rdiv__**(*self*, *other*)

`--rmul--(self, other)``--rpow--(self, other)``--rsub--(self, other)``--rtruediv--(self, other)``--sub--(self, other)``--truediv--(self, other)`***Inherited from object***`--delattr--()`, `--format--()`, `--getattr--()`, `--hash--()`, `--new--()`, `--reduce--()`, `--reduce_ex--()`, `--repr--()`, `--setattr--()`, `--sizeof--()`, `--subclasshook--()`**98.8.2 Properties**

Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

98.9 Class `operator1`

98.9.1 Methods

`__init__(self, operator, operand)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` extit(inherited documentation)

`__str__(self, env={})`

Env is optional. Anything defined in `env` is used to resolve references, but the outcome may still contain undefined variables, which are printed as names.

Overrides: `object.__str__` extit(inherited documentation)

`__repr__(self, env={})`

Env is optional. Anything defined in `env` is used to resolve references, but the outcome may still contain undefined variables, which are printed as names.

Overrides: `object.__repr__` extit(inherited documentation)

`eval(self, env={})`

This returns an Expression.

Overrides: `gmisclib.parse_tree.number.output_mixin.eval` extit(inherited documentation)

`variables(self, env={})`

Any definitions in `env` are used to resolve otherwise undefined names. This returns a Set containing all undefined variable names.

Overrides: `gmisclib.parse_tree.number.output_mixin.variables` extit(inherited documentation)

`indices(self, variable, env={})`

Overrides: `gmisclib.parse_tree.number.output_mixin.indices`

`debug(self)`

Overrides: `gmisclib.parse_tree.number.output_mixin.debug`

`__abs__(self)`

`--add__(self, other)``--div__(self, other)``--mul__(self, other)``--neg__(self)``--pos__(self)``--pow__(self, other)``--radd__(self, other)``--rdiv__(self, other)``--rmul__(self, other)``--rpow__(self, other)``--rsub__(self, other)``--rtruediv__(self, other)``--sub__(self, other)``--truediv__(self, other)`*Inherited from `gmisclib.parse_tree_number._operator`*`priority()`*Inherited from `object`*`--delattr__(), --format__(), --getattr__(), --hash__(), --new__(), --reduce__(), --reduce_ex__(),
--setattr__(), --sizeof__(), --subclasshook__()`

98.9.2 Properties

Name	Description
<i>Inherited from <code>object</code></i>	

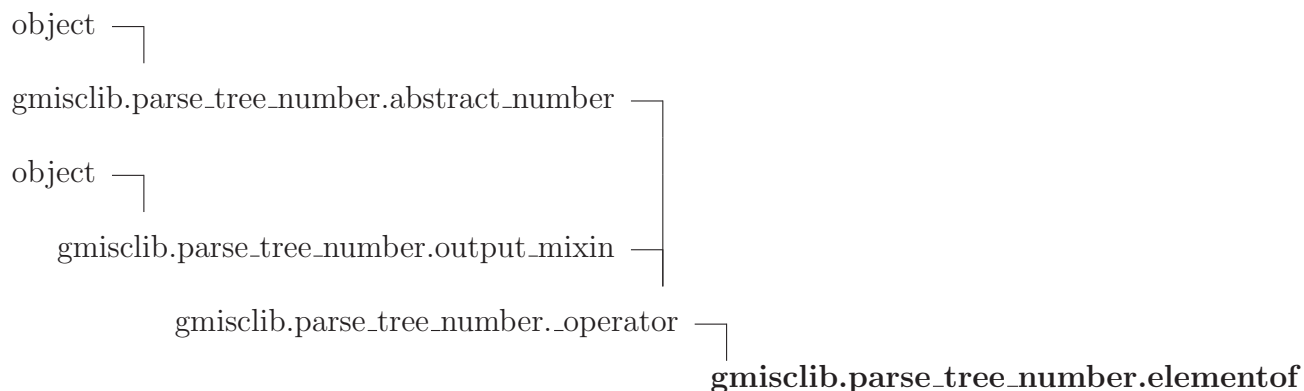
continued on next page

Name	Description
<code>__class__</code>	

98.9.3 Class Variables

Name	Description
Printable	Value: { <code>'U-'</code> : <code>'-'</code> , <code>'abs'</code> : <code>'abs'</code> , <code>'cos'</code> : <code>'math.cos'</code> , <code>'exp'</code> : <code>'math...</code>
<i>Inherited from gmisclib.parse_tree_number._operator</i>	
Priority	

98.10 Class elementof



98.10.1 Methods

`__init__`(*self*, *array*, *index*)

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` extit(inherited documentation)

`__str__`(*self*, *env*={})

`Env` is optional. Anything defined in `env` is used to resolve references, but the outcome may still contain undefined variables, which are printed as names.

Overrides: `object.__str__` extit(inherited documentation)

```
--repr--(self, env={})
```

Env is optional. Anything defined in env is used to resolve references, but the outcome may still contain undefined variables, which are printed as names.

Overrides: object.__repr__ exitit(inherited documentation)

```
variables(self, env={})
```

Any definitions in env are used to resolve otherwise undefined names. This returns a Set containing all undefined variable names.

Overrides: gmisclib.parse_tree_number.output_mixin.variables exitit(inherited documentation)

```
indices(self, variable, env={})
```

Overrides: gmisclib.parse_tree_number.output_mixin.indices

```
debug(self)
```

Overrides: gmisclib.parse_tree_number.output_mixin.debug

```
--abs--(self)
```

```
--add--(self, other)
```

```
--div--(self, other)
```

```
--mul--(self, other)
```

```
--neg--(self)
```

```
--pos--(self)
```

```
--pow--(self, other)
```

```
--radd--(self, other)
```

```
--rdiv--(self, other)
```

```
--rmul--(self, other)
```

<code>--rpow--(<i>self</i>, <i>other</i>)</code>
--

<code>--rsub--(<i>self</i>, <i>other</i>)</code>
--

<code>--rtruediv--(<i>self</i>, <i>other</i>)</code>
--

<code>--sub--(<i>self</i>, <i>other</i>)</code>

<code>--truediv--(<i>self</i>, <i>other</i>)</code>

<code>eval(<i>self</i>, <i>env</i>={})</code>

This returns an Expression.

Inherited from gmisclib.parse_tree_number._operator

priority()

Inherited from object

`--delattr--()`, `--format--()`, `--getattr--()`, `--hash--()`, `--new--()`, `--reduce--()`, `--reduce_ex--()`,
`--setattr--()`, `--sizeof--()`, `--subclasshook--()`

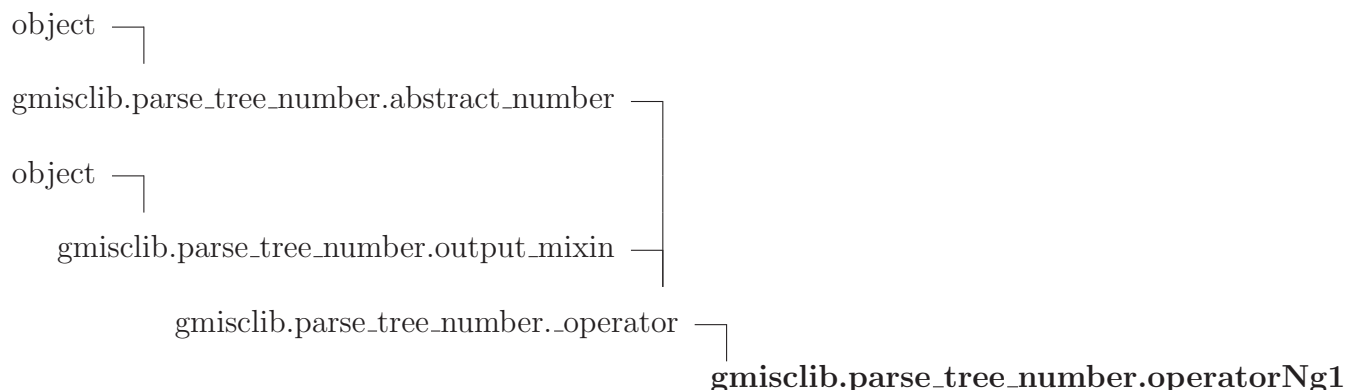
98.10.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

98.10.3 Class Variables

Name	Description
<i>Inherited from gmisclib.parse_tree_number._operator</i>	
Priority	

98.11 Class `operatorNg1`



98.11.1 Methods

`__init__(self, operator, operands)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` `exitit`(inherited documentation)

`more(self, items)`

`__str__(self, env={})`

Env is optional. Anything defined in `env` is used to resolve references, but the outcome may still contain undefined variables, which are printed as names.

Overrides: `object.__str__` `exitit`(inherited documentation)

`__repr__(self, env={})`

Env is optional. Anything defined in `env` is used to resolve references, but the outcome may still contain undefined variables, which are printed as names.

Overrides: `object.__repr__` `exitit`(inherited documentation)

`variables(self, env={})`

Any definitions in `env` are used to resolve otherwise undefined names. This returns a Set containing all undefined variable names.

Overrides: `gmisclib.parse_tree_number.output_mixin.variables` `exitit`(inherited documentation)

indices(*self*, *variable*, *env*={})Overrides: `gmisclib.parse_tree.number.output_mixin.indices`**debug**(*self*)Overrides: `gmisclib.parse_tree.number.output_mixin.debug`**__abs__**(*self*)**__add__**(*self*, *other*)**__div__**(*self*, *other*)**__mul__**(*self*, *other*)**__neg__**(*self*)**__pos__**(*self*)**__pow__**(*self*, *other*)**__radd__**(*self*, *other*)**__rdiv__**(*self*, *other*)**__rmul__**(*self*, *other*)**__rpow__**(*self*, *other*)**__rsub__**(*self*, *other*)**__rtruediv__**(*self*, *other*)**__sub__**(*self*, *other*)**__truediv__**(*self*, *other*)**eval**(*self*, *env*={})

This returns an Expression.

Inherited from `gmisclib.parse_tree_number._operator`

`priority()`

Inherited from object

`__delattr__()`, `__format__()`, `__getattribute__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__setattr__()`, `__sizeof__()`, `__subclasshook__()`

98.11.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

98.11.3 Class Variables

Name	Description
<i>Inherited from <code>gmisclib.parse_tree_number._operator</code></i>	
Priority	

99 Module *gmisclib.permute*

Find different permutations of an array.

99.1 Functions

next (<i>a</i> , <i>n</i>)

Finds the first valid permutation of <i>n</i> items after <i>a</i> . A valid permutation is an array of length <i>n</i> which contains values from 0 to <i>n</i> -1, once each. This function returns the lexicographically next permutation; it's input need not be a valid permutation, but all the entries of <i>a</i> should be in the range 0 to <i>n</i> -1, inclusive.

factorial (<i>n</i>)

permutations (<i>n</i>)

Returns an array of all the <i>n</i> ! permutations of <i>n</i> objects.
--

99.2 Variables

Name	Description
<code>__package__</code>	Value: <code>'gmisclib'</code>

100 Module `gmisclib.pylab.oneaxis`

You can control the axis tick and grid properties

100.1 Functions

plot_oneaxis (<i>dlist</i>)

<i>dlist</i> is a list of (label, (ctr, uerr, derr)).

101 Module `gmisclib.pylab_starplot`

This makes a little 'star' of error bars, using `pylab/matplotlib`.

101.1 Functions

`grey_out(c)`

`star(cx, cy, lch, lcolormap=None, mcolormap=None, dimmer=grey_out, **kwargs)`

This makes a star of error bars.

Parameters

cx: x-coordinate of the star center
cy: y-coordinate of the star center
lch: list of error bars to create about the center.
(type=list(errorbar).)

`plot(xlch, **kwargs)`

101.2 Variables

Name	Description
Grey	Value: 0.7, 0.7, 0.7

101.3 Class `errorbar_maker`

object └─ `gmisclib.pylab_starplot.errorbar_maker`

This is a class that makes error bars, tilted at various angles.

101.3.1 Methods

```
__init__(self, cx, cy, n, **kwargs)
```

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

```
make(self, i, xl, xc, xh, **kwargs)
```

Inherited from object

__delattr__(), __format__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
__repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

101.3.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

101.4 Class errorbar

```
object └─ gmisclib.pylab.starplot.errorbar
```

101.4.1 Methods

```
__init__(self, l, c, h, dim=False, sortorder=None, **kwargs)
```

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

```
__mul__(self, other)
```

```
__cmp__(self, other)
```

Inherited from object

__delattr__(), __format__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
__repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

101.4.2 Properties

Name	Description
<i>Inherited from object</i> <code>__class__</code>	

102 Module `gmisclib.read_dicom`

102.1 Functions

<code>read_header(f)</code>

<code>read_body(f)</code>

<code>read_imgs(f)</code>

102.2 Variables

Name	Description
<code>__package__</code>	Value: <code>'gmisclib'</code>

102.3 Class `img_with_mx`

object —
 `gmisclib.read_dicom.img_with_mx`

This is a way of reading in a 2-D image when you don't know the final size. It keeps a larger image around and keeps track of the area that has been set.

102.3.1 Methods

<code>__init__(self, i, j)</code>

Create an image that contains pixel (i,j).
--

Overrides: <code>object.__init__</code>

<code>resize(self, inxt, jnxt)</code>

<code>set(self, i, j, val)</code>

Set a pixel, expanding the allocated space as needed.

<code>get(self)</code>

Get the image, trimming it to show the area that has been set.
--

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

102.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

103 Module *gmisclib.resample*

Resample a time series by a factor *r*. I.e. sample more densely ($r < 1$) or less densely ($r > 1$). Also see *tsops.py*.

103.1 Functions

<code>simple(<i>a</i>, <i>r</i>, <i>direction</i>=0)</code>

<code>compare(<i>a</i>, <i>b</i>)</code>
--

<code>test()</code>

103.2 Variables

Name	Description
<code>--package--</code>	Value: <code>'gmisclib'</code>

104 Module *gmisclib.robust_multivariate*

Does a robust estimate of covariance. It doesn't converge, unfortunately, for vectors of dimension higher than 1.

104.1 Functions

cov_wt (<i>veclist</i> , <i>weights</i>)

integrate_guts (<i>xlow</i> , <i>xhigh</i> , <i>fcn</i> , <i>c</i> , <i>n</i>)

integrate (<i>xlow</i> , <i>xhigh</i> , <i>fcn</i> , <i>c=None</i> , <i>nstart=5</i> , <i>tol=0.0001</i>)
--

A generally useful function, does the integral of <i>fcn</i> (<i>x</i> , <i>c</i>) from <i>xlow</i> to <i>xhigh</i> .

wtfunc (<i>q</i> , <i>E</i>)

wtdim (<i>q</i> , <i>Edim</i>)

covariance (<i>veclist</i> , <i>emax=1.0</i>)
--

<i>Veclist</i> [<i>i</i>] is a vector.
--

104.2 Variables

Name	Description
<code>--package--</code>	Value: 'gmisclib'

105 Module *gmisclib.root*

Solve a 1-dimensional equation to find the roots. This does alternating secant and bisection steps. It will work for any function, discontinuous or not. For a nasty function, it might be a factor of 2 slower than bisection, but for a nearly-linear function, it can be much faster than bisection.

105.1 Functions

root(*f, xl, xh, p, epsx, epsf*)

Find a root of an equation.

Parameters

f: is the function to solve, called as $f(x, p)$.

p: is arbitrary data for **f**. This function assumes there is known to be a root in $[xl, xh]$ and that $f(xl, p)$ has a different sign from $f(xh, p)$. It finds it within a region of width *epsx*, or at least finds an x such that $-epsf < f(x, p) < epsf$.

Return Value

a root in the interval

(*type=float*)

test()

iroot(*y, xl, xh, epsy=0.0*)

Find a zero in an array *y*. This function assumes there is known to be a root in $[xl, xh]$. It returns a real-number index into the array which linearly interpolates to zero.

testi()

105.2 Variables

Name	Description
<code>--package--</code>	Value: None

106 Module `gmisclib.rubber_array`

This acts like a dictionary, but indexed by integers and stored rather more efficiently, at least in terms of memory. It is designed for very sparse arrays.

106.1 Variables

Name	Description
<code>__package__</code>	Value: <code>'gmisclib'</code>

106.2 Class `ext_block`

object └─ `gmisclib.rubber_array.ext_block`

106.2.1 Methods

```
__init__(self, n, vtype=<type 'numpy.float64'>)
```

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` `extit`(inherited documentation)

```
set(self, k, v)
```

```
get(self, k, IdxErrArg)
```

```
contains(self, value)
```

Inherited from `object`

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

106.2.2 Properties

Name	Description
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

106.3 Class `extensible_array`



106.3.1 Methods

`--init--(self, bsz=300)`

`x.--init--(...)` initializes `x`; see `x.--class--.--doc--` for signature

Overrides: `object.--init--` `exitit`(inherited documentation)

`--len--(self)`

`--setitem--(self, k, v)`

Doesn't do slice objects.

`--getitem--(self, k)`

`unset(self, key)`

`--contains--(self, item)`

`--iter--(self)`

Inherited from object

`--delattr--()`, `--format--()`, `--getattr--()`, `--hash--()`, `--new--()`, `--reduce--()`, `--reduce_ex--()`,
`--repr--()`, `--setattr--()`, `--sizeof--()`, `--str--()`, `--subclasshook--()`

106.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

107 Module gmisclib.s_lin_fit

Fit a plane to some data.

107.1 Functions

plane(*data*, *wt*=None)

Fit a plane to some data. where the fitting function is $f = c[0]*1 + c[1]*dep[0] + c[2]*dep[1] + \dots$ where *c* is the array of coefficients. The length of *c* is equal to the number of dependent variables in each datum.

Parameters

data: [(independent, dependent, dependent...), ...]

Return Value

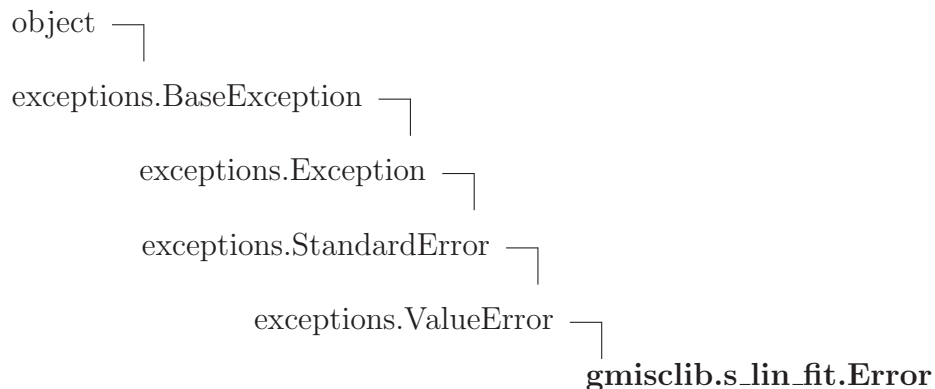
(*c*, *opt*, *resid*, *rank*, *sv*), where *bestfit* is the best fit to the data (i.e., the values of *f*), *resid* is the residual (a single float number), *rank* the rank of the fit (int), and *sv* is the array of the singular values.

test()

107.2 Variables

Name	Description
<code>__package__</code>	Value: 'gmisclib'

107.3 Class Error



107.3.1 Methods

`__init__`(*self*, *s*)

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` `extit`(inherited documentation)

Inherited from `exceptions.ValueError``__new__()`***Inherited from `exceptions.BaseException`***

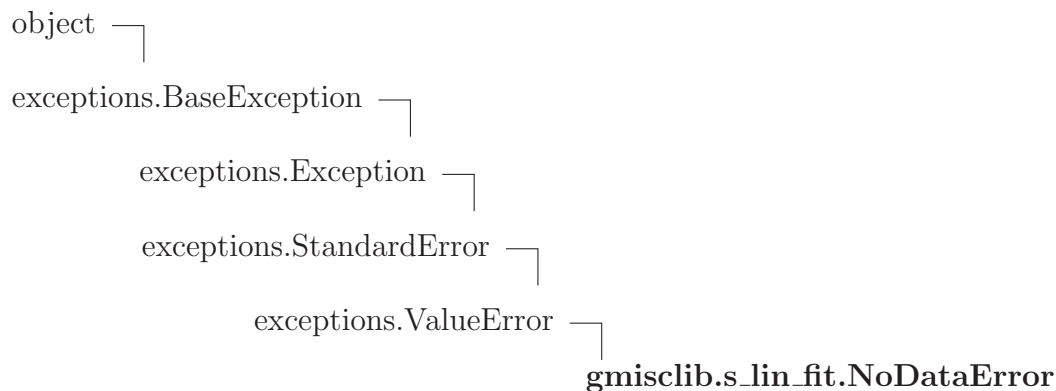
`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

Inherited from `object`

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

107.3.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
<code>args</code> , <code>message</code>	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

107.4 Class `NoDataError`

107.4.1 Methods

`__init__(self, s)`
`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature
 Overrides: `object.__init__` extit(inherited documentation)

Inherited from exceptions.ValueError

`__new__()`

Inherited from exceptions.BaseException

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

Inherited from object

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

107.4.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
<code>args</code> , <code>message</code>	
<i>Inherited from object</i>	
<code>__class__</code>	

108 Module *gmisclib.sbd_array*

108.1 Functions

<code>symmetrize</code> (<i>matrix</i> , <i>direction</i>)

<code>solve</code> (<i>a</i> , <i>b0</i>)
--

solves $a \cdot x = b$, where <i>a</i> is class <i>sbd</i> : symmetric, positive definite, and band-diagonal. Note that this destroys the contents of <i>a</i> .

<code>multiply</code> (<i>a</i> , <i>x</i>)
--

Calculates $a \cdot x$, where <i>a</i> is class <i>sbd</i> : symmetric, positive definite, and band-diagonal.
--

<code>test1</code> ()

<code>err</code> (<i>a</i> , <i>b</i>)

<code>test2</code> ()

<code>testm</code> ()

<code>testa</code> ()

<code>testbdi</code> ()

108.2 Variables

Name	Description
<code>--package--</code>	Value: 'gmisclib'

108.3 Class *sbd*

108.3.1 Methods

`--init--`(*self*, *sz*, *bw*, *_data=None*)

Constructs a symmetric band-diagonal matrix of $sz*sz$, with a nonzero bandwidth of *bw*. $bw==0$ corresponds to a diagonal matrix.

`--copy--`(*self*)

Copy the data, not just the data description.

`--getitem--`(*self*, *key*)

Pulls an element of the array. Key is a 2-tuple that specifies the 'virtual' position in the array (i.e., as if the array were a full square matrix, rather than a band diagonal symmetric matrix).

`--setitem--`(*self*, *key*, *value*)

`increment`(*self*, *key*, *delta*)

Increment a single value (and its symmetric partner, if off the diagonal).

`bd_increment`(*self*, *key*, *delta*)

Block diagonal increment.

`--str--`(*self*)

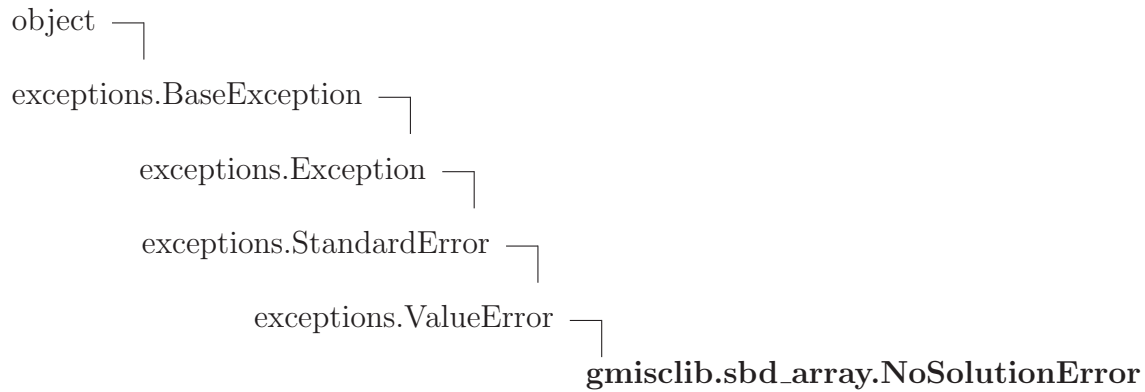
`--repr--`(*self*)

`--pow--`(*self*, *other*)

`--getslice--`(*self*, *i*, *j*)

This is copy semantics, not shared reference.

108.4 Class NoSolutionError



108.4.1 Methods

`--init--`(*self*, *s*)

`x.--init--(...)` initializes `x`; see `x.--class--.--doc--` for signature

Overrides: `object.--init--` `exitit`(inherited documentation)

Inherited from exceptions.ValueError

`--new--`()

Inherited from exceptions.BaseException

`--delattr--`(), `--getattr--`(), `--getitem--`(), `--getslice--`(), `--reduce--`(), `--repr--`(),
`--setattr--`(), `--setstate--`(), `--str--`(), `--unicode--`()

Inherited from object

`--format--`(), `--hash--`(), `--reduce.ex--`(), `--sizeof--`(), `--subclasshook--`()

108.4.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
args, message	
<i>Inherited from object</i>	
<code>--class--</code>	

109 Module gmisclib.segmentfile

This parses a '.in' file from xwaves. The '.in' files show segmentation of an utterance. `segmentfile.parse(s)` takes a list of lines from such a file, and returns three things: * the title of the file (typically the utterance) * a list of groups (a group is typically a word) * a list of segments.

Groups include a list of the segments of which they are made, along with a name and an index. Segments contain a phoneme, a start/end time, an index, and the group to which they belong.

109.1 Functions

parse(*l*)

Parses a list of lines from an ESPS Xmark file (.in), and returns a tuple (title, list of groups, list of segments).

read(*f*)

Read a ESPS Xmark (.in) file, and return information. See `parse()`.

test()

109.2 Variables

Name	Description
<code>__package__</code>	Value: None

109.3 Class segment

109.3.1 Methods

__init__(*self*, *st*, *en*, *phn*, *si*)

setgroup(*self*, *g*)

__str__(*self*)

```
--repr--(self)
```

109.4 Class group

109.4.1 Methods

```
--init--(self, gi, gn)
```

```
add(self, seg)
```

```
--str--(self)
```

```
--repr--(self)
```

110 Module `gmisclib.sharp.energy`

110.1 Functions

<code>complex_median</code> (<i>x</i>)

<code>one_median</code> (<i>binw</i> , <i>center</i> , <i>v</i> , <i>nbins</i> =None)

Local power in a signal. <i>dt</i> and <i>binw</i> are measured in bins.
--

<code>stepped_median</code> (<i>dt</i> , <i>binw</i> , <i>v</i> , <i>nbins</i> =None)

Local power in a signal. <i>dt</i> and <i>binw</i> are measured in bins.
--

<code>pwr</code> (<i>dt</i> , <i>binw</i> , <i>v</i> , <i>nbins</i> =3)

111 Module *gmisclib.solve_sum_abs*

Solves various equations involving minimizing the sum of absolute values of things.

111.1 Functions

`solve1`(*y0*, *y1*)

We minimize the equation $\text{sum_over_i}(\text{abs}(y0_i \cdot (1-x) + y1_i \cdot x))$ to find the best *x*. Returns *x*. I don't know if this is strictly correct.

`solve_fit`(*x*, *y*, *epsx*=1e-07, *epsm*=1e-07)

Solves for the line $y_{\text{hat}} = m \cdot x + b$ that minimizes $\text{sum}(\text{abs}(y - y_{\text{hat}}))$. In other words, it's a fit to a straight line, but a highly robust fit that will not be disturbed by outliers. The algorithm is from Numerical Recipes, Volume 2.

Parameters

`epsx`: a tolerance used in the iterative solution. Eps is roughly the required accuracy of the x-intercept of the solution, expressed as a fraction of the x-range of the data.

(*type*=*float*)

`epsm`: a tolerance used in the iterative solution. Roughly, it is the accuracy of *m* in the solution. Note the `tangent()` call!

(*type*=*float*)

Return Value

(*mhat*, *bhat*), where $m_{\text{hat}} \cdot x + b_{\text{hat}}$ is the best fit to the data.

(*type*=*tuple(float, float)*)

`solve_fit_wt`(*x*, *y*, *wt*, *epsx*=1e-07, *epsm*=1e-07)

Solves for the line $yhat = m*x + b$ that minimizes $sum(abs(y - yhat))$. In other words, it's a fit to a straight line, but a highly robust fit that will not be disturbed by outliers. The algorithm is from Numerical Recipes, Volume 2.

Parameters

`epsx`: a tolerance used in the iterative solution. Eps is roughly the required accuracy of the x-intercept of the solution, expressed as a fraction of the x-range of the data.

(*type*=*float*)

`epsm`: a tolerance used in the iterative solution. Roughly, it is the accuracy of *m* in the solution. Note the `tangent()` call!

(*type*=*float*)

Return Value

(*mhat*, *bhat*), where $mhat*x + bhat$ is the best fit to the data.

(*type*=*tuple(float, float)*)

`test1`()

`test2`()

111.2 Variables

Name	Description
<code>__package__</code>	Value: <code>'gmisclib'</code>

112 Module *gmislib.spread_jobs*

A module that starts a bunch of subprocesses and distributes work amongst them, then collects the results.

Subprocesses must follow a protocol: they must listen for commands on the standard input (commands are encoded with `cPickle`), and they must produce `cPickle`d tuples on their standard output. NOTE: THEY CANNOT PRINT ANYTHING ELSE! (But it's OK for subprocesses to produce stuff on the standard error output.)

PROTOCOL:

1. Execcing state: `spread_jobs` execs a group of subprocesses. You have full control of the argument list.
2. Preparation state: `spread_jobs` will send a list of `cPickle`d things, one at a time to the subprocess. No explicit terminator is added, so the subprocess must either know how many things are coming or the list should contain some terminator. (E.g. the last item of the list could be `None`, and the subprocess would wait for it.) In this state, the subprocess must not produce anything on the standard output.
3. Running state: `spread_jobs` will send one `cPickle`d thing to the subprocess and then wait for a `cPickle`d tuple to come back. The tuple has three items: (0) is an arbitrary payload, (1) a list of strings to be printed to the main processes' `sys.stdout`, and (2) another list of strings to be printed to the main processes' `sys.stderr`. The arbitrary payload is put on a list and (when all the subprocesses are finished) handed to whoever called `spread_jobs.main2`.
4. Shutdown state: The subprocess can produce anything it wants. This will be collected up and returned to the caller of `spread_jobs.main2`.

You can use this to run certain normal linux commands by not sending anything in the preparatory state or the running state. You will then be handed the standard output as a list of strings.

Normally, however, the action happens in the running state.

Normally, the subprocess looks much like this:

```
import cPickle
while True:
    try:
        control = cPickle.load(stdin)
    except EOFError:
        break
    d, so, se = compute(control)
    cPickle.dump((d, so, se), stdout, CP.HIGHEST_PROTOCOL)
    stdout.flush()
stdout.close()
```

112.1 Functions

main2(*todo*, *list_of_args*, *stdin*=None, *verbose*=False, *timing_callback*=None)

Pass a bunch of work to other processes.

Parameters

stdin: a list of stuff to send to the other processes before the computation is properly commenced.

(*type*=*list(whatever)*)

todo: a list of work to do

(*type*=*list(whatever)*)

list_of_args: (*type*=*list(list(str))*)

Return Value

A 2-tuple. The first item is a list of the results produced by the other processes. Items in the returned list correspond to items in the *todo* list. These are the stuff that comes out, pickled, on the standard output after each chunk of data is fed into the standard input. The second item is a list of the remaining outputs, as read by `file.readlines()`; these are one per process.

(*type*=*tuple(list(whatever), list(list(str)))*)

main(*todo*, *list_of_args*, *stdin*=None, *verbose*=False, *timing_callback*=None)

Pass a bunch of work to other processes.

Parameters

stdin: a list of stuff to send to the other processes before the computation is properly commenced.

(*type*=*list(whatever)*)

todo: a list of work to do

(*type*=*list(whatever)*)

list_of_args: (*type*=*list(list(str))*)

Return Value

a list of the results produced by the other processes. Items in the returned list correspond to items in the *todo* list.

(*type*=*list(whatever)*)

replace(*list_of_lists*, **fr*)

Replace(*list_of_lists*, *pat*, *length*, *replacement*)

```
append(list_of_lists, *a)
```

```
test_worker()
```

```
test()
```

```
one_shot_test(input)
```

112.2 Variables

Name	Description
<code>--package--</code>	Value: 'gmisclib'

112.3 Class *notComputed*

object └─ **gmisclib.spread_jobs.notComputed**

A singleton marker for values that haven't been computed.

112.3.1 Methods

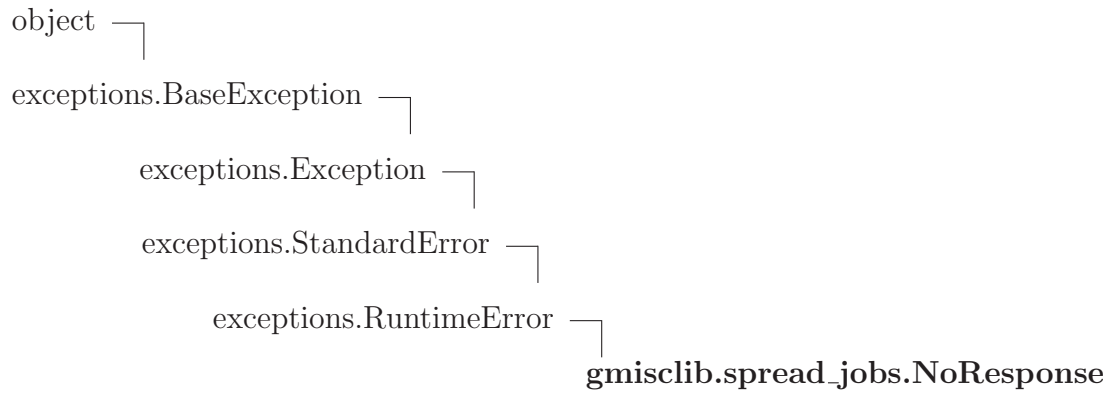
Inherited from object

```
--delattr--(), --format--(), --getattrattribute--(), --hash--(), --init--(), --new--(), --reduce--(),
--reduce_ex--(), --repr--(), --setattr--(), --sizeof--(), --str--(), --subclasshook--()
```

112.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

112.4 Class NoResponse



112.4.1 Methods

`__init__(self, *s)`
`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature
 Overrides: `object.__init__` extit(inherited documentation)

Inherited from exceptions.RuntimeError

`__new__()`

Inherited from exceptions.BaseException

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

Inherited from object

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

112.4.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
<code>args</code> , <code>message</code>	
<i>Inherited from object</i>	
<code>__class__</code>	

112.5 Class RemoteException



An exception that corresponds to one raised by a subprocess. This is raised in the parent process.

112.5.1 Methods

```
__init__(self, *s)
```

x.**__init__**(...) initializes *x*; see *x*.**__class__**.**__doc__** for signature
 Overrides: *object*.**__init__** *exitit*(inherited documentation)

```
__repr__(self)
```

repr(*x*)
 Overrides: *object*.**__repr__** *exitit*(inherited documentation)

```
raise_me(self)
```

Inherited from exceptions.Exception

```
__new__()
```

Inherited from exceptions.BaseException

```
__delattr__(), __getattribute__(), __getitem__(), __getslice__(), __reduce__(), __setattr__(),  

__setstate__(), __str__(), __unicode__()
```

Inherited from object

```
__format__(), __hash__(), __reduce_ex__(), __sizeof__(), __subclasshook__()
```

112.5.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	

continued on next page

Name	Description
args, message	
<i>Inherited from object</i>	
__class__	

112.6 Class `unpickled_pseudofile`

StringIO.StringIO —
 gmisclib.spread_jobs.unpickled_pseudofile

For testing.

112.6.1 Methods

__init__(self)
 Overrides: StringIO.StringIO.__init__

close(self)
 Free the memory buffer.
 Overrides: StringIO.StringIO.close exitit(inherited documentation)

Inherited from StringIO.StringIO

`__iter__()`, `flush()`, `getvalue()`, `isatty()`, `next()`, `read()`, `readline()`, `readlines()`, `seek()`,
`tell()`, `truncate()`, `write()`, `writelines()`

113 Module *gmisclib.stats*

Some of these functions, specifically `f_value()`, `fprob()`, `betai()`, and `betacf()`, are taken from `stats.py` "A collection of basic statistical functions for python." by Gary Strangman. They are copyright 1999-2007 Gary Strangman; All Rights Reserved and released under the MIT license.

The rest is copyright Greg Kochanski 2009.

113.1 Functions

f_value(*ER*, *EF*, *dfR*, *dfF*)

Returns an F-statistic given the following:

Parameters

- ER:** error associated with the null hypothesis (the Restricted model)
- EF:** error associated with the alternate hypothesis (the Full model)
- dfR:** degrees of freedom the Restricted model
- dfF:** degrees of freedom associated with the Full model

Return Value

f-statistic (not the probability)
(*type=float*)

fprob(*dfnum*, *dfden*, *F*)

Returns the (1-tailed) significance level (p-value) of an F statistic given the degrees of freedom for the numerator (*dfR-dfF*) and the degrees of freedom for the denominator (*dfF*).

Usage: `lfprob(dfnum, dfden, F)` where usually *dfnum=dfbn*, *dfden=dfwn*

betai(*a*, *b*, *x*)

Returns the incomplete beta function:

$$I\text{-sub-x}(a,b) = 1/B(a,b) * (\text{Integral}(0,x) \text{ of } t^{(a-1)}(1-t)^{(b-1)} dt)$$

where $a \geq 0, b > 0$ and $B(a,b) = G(a)G(b)/(G(a+b))$ where $G(a)$ is the gamma function of a . The continued fraction formulation is implemented here, using the `betacf` function. (Adapted from: Numerical Recipes in C.)

Usage: `betai(a,b,x)`

betacf(*a*, *b*, *x*)

This function evaluates the continued fraction form of the incomplete Beta function, betai. (Adapted from: Numerical Recipes in C.)

Usage: betacf(a,b,x)

gammaln(*x*)

Returns the gamma function of xx. $\Gamma(z) = \int_0^\infty t^{z-1} \exp(-t) dt$. (Adapted from: Numerical Recipes in C. via code Copyright (c) 1999-2000 Gary Strangman and released under the LGPL.)

test_fprob(*dfnum*, *dfden*)

t_value(*ndof*, *p2sided*=0.99)

113.2 Variables

Name	Description
t_Table	Value: [(1, 0.5, 1.0), (1, 0.95, 12.71), (1, 0.98, 31.82), (1, 0...
__package__	Value: 'gmisclib'

114 Module *gmisclib.synch_demod*

114.1 Functions

pwr_guts(*v*, *dt*, *binw*, *w*, *K*)

Local power in a signal. *dt* and *binw* are measured in terms of the sampling interval of *v*. *w* is the bandwidth of the filter. *K* is the order of the prolate spheroidal fcn.

pwr(*v*, *dt*, *bandwidth*, *K*)

expand(*r*, *a*)

pwrs(*dt*, *harmonics*, *f0*, *signal*, *sig_dt*, *bandwidth*)

This is a classical synchronous demodulator, with DPSS windowing to get maximally steep skirts and out-of-band rejection.

dt=output power measurement interval. *harmonics* = [which harmonics to measure (multiples of *f0*)] *f0* = frequency as a function of time (measured at intervals of *dt*) *signal* = input signal to be demodulated *sig_dt* = sampling interval of signal.

one_ampl(*harmonics*, *phase*, *signal*, *binw*, *nbin=None*)

Everything measured in samples. Phase and signal have same sampling rate.

stepped_ampl(*dt*, *harmonics*, *f0*, *signal*, *sig_dt*, *subwindow=0.010*, *nsubw=4*)

signal -> amplitudes in specified harmonics *dt*=time interval of *f0* samples = time interval of *ampl* measurements.

test0()

test1()

115 Module `gmislib.threaded_io`

This is designed to let you do asynchronous I/O conveniently.

115.1 Functions

to_be_joined(*tlist*)

Pick a ripe thread for joining.

Parameters

tlist: a list of threads.

(*type=list(threading.Thread)*)

Return Value

a thread that is either finished and ready to be joined, or (if none are ready yet) the oldest thread.

(*type=threading.Thread*)

check_n_raise(*x*)

pairmap(*fn, inlist, *args, **kwargs*)

ncpu()

Count the number of processors on this machine.

Return Value

the integer number of cores that the computer has.

(*type=int*)

testmap()

115.2 Variables

Name	Description
<code>--package--</code>	Value: <code>'gmislib'</code>

115.3 Class `CaughtException`

object └─
 `gmisclib.threaded_io.CaughtException`

115.3.1 Methods

`__init__(self, x)`

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` `exitit`(inherited documentation)

`__repr__(self)`

`repr(x)`

Overrides: `object.__repr__` `exitit`(inherited documentation)

`reraise(self)`

Inherited from `object`

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

115.3.2 Properties

Name	Description
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

115.4 Class `threading_with_result`

object └─
 threading.Verbose └─
 threading.Thread └─
 `gmisclib.threaded_io.threading_with_result`

115.4.1 Methods

```
__init__(self, tag=None, group=None, target=None, name=None, args=(),
kwargs={})
```

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` `exitit`(inherited documentation)

```
run(self)
```

Overrides: `threading.Thread.run`

```
get(self)
```

Inherited from `threading.Thread`

`__repr__()`, `getName()`, `isAlive()`, `isDaemon()`, `is_alive()`, `join()`, `setDaemon()`, `setName()`, `start()`

Inherited from `object`

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

115.4.2 Properties

Name	Description
<i>Inherited from <code>threading.Thread</code></i>	
<code>daemon</code> , <code>ident</code> , <code>name</code>	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

115.5 Class `Thread_poolW`

```
object └─
          gmislib.threaded_io.Thread_poolW
```

Here, you have a pool of `n` threads. You can call `x.start(function, args, kwargs)` to start something running (perhaps a write to a file), and then go off and compute something else while the I/O completes. If you need to wait for completion, call `x.join()`.

This class is not designed to return values from the function.

115.5.1 Methods

```
__init__(self, n=None)
```

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` `exitit`(inherited documentation)

```
start(self, fn, args=None, kwargs=None)
```

```
join(self)
```

Inherited from object

`__delattr__()`, `__format__()`, `__getattr__()`, `__hash__()`, `__new__()`, `__reduce__()`, `__reduce_ex__()`,
`__repr__()`, `__setattr__()`, `__sizeof__()`, `__str__()`, `__subclasshook__()`

115.5.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

115.6 Class `Thread_poolW`

```
object └─
          gmislib.threaded_io.Thread_poolW
```

Here, you have a pool of `n` threads. You can call `x.start(function, args, kwargs)` to start something running (perhaps a write to a file), and then go off and compute something else while the I/O completes. If you need to wait for completion, call `x.join()`.

This class is not designed to return values from the function.

115.6.1 Methods

```
__init__(self, n=None)
```

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` `exitit`(inherited documentation)

```
start(self, fn, args=None, kwargs=None)
```

```
join(self)
```

Inherited from object

```
__delattr__(), __format__(), __getattr__(), __hash__(), __new__(), __reduce__(), __reduce_ex__(),
__repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()
```

115.6.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

115.7 Class Thread_poolR

object └─ **gmisclib.threaded_io.Thread_poolR**

Here, you have a pool of n threads. You can call **x.start(function, args, kwargs)** to start something running (perhaps a write to a file), and then go off and compute something else while the I/O completes. If you need to wait for completion, call **x.join()**.

This class is designed to return values from the function via **get()** or **getany()**.

115.7.1 Methods

```
__init__(self, n=None)
```

x.__init__(...) initializes x; see **x.__class__.__doc__** for signature

Overrides: **object.__init__** **exitit**(inherited documentation)

```
start(self, tag, fn, args=None, kwargs=None)
```


getany (<i>self</i>)

Return Value

the first available answer, as a (tag, value) pair. The tag was set in the call to start and the value is the result of the computation. If the computation raised an exception, getany will return a CaughtException object instead.
--

(<i>type</i> =a tuple(<i>whatever</i> , <i>whatever</i>) pair from a computation or a CaughtException)

get (<i>self</i> , <i>tag</i>)

Return Value

whatever or a CaughtException

has_answer (<i>self</i>)

Is there an answer ready?

loaded (<i>self</i>)

Have we loaded up all the processors?

Inherited from object

`--delattr--()`, `--format--()`, `--getattr--()`, `--hash--()`, `--new--()`, `--reduce--()`, `--reduce_ex--()`, `--repr--()`, `--setattr--()`, `--sizeof--()`, `--str--()`, `--subclasshook--()`

115.7.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

116 Module *gmisclib.tsops*

Do mathematical operations on time series, where the two operands don't necessarily have the same sampling times. It finds a common sampling time and sampling interval, then interpolates as necessary to bring the data onto the common time axis.

116.1 Functions

```
time(datasets, start=None, end=None)
```

```
time2(a, b)
```

```
test_fig()
```

```
interp_fill(a, t, fill)
```

```
interpN_fill(a, t, fill)
```

```
interp(a, t)
```

Interpolate to a specified time axis. This does a linear interpolation.

Parameters

a: data to be interpolated (a time series)

(*type*=*gpkimgclass.gpk_img*)

t: (*type*=*an array of times.*)

Return Value

data interpolated onto the specified time values.

(*type*=*numpy array.*)

```
test_interp1()
```

```
test_interp2()
```

```
test_interp3()
```

interpN(*a*, *t*)

Interpolate to a specified time axis via nearest-neighbor interpolation. *A* is a *gpkimgclass*, and *t* is an array of times. Returns a Numeric array, not a *gpkimgclass*.

common(*data_sets*, *start*=None, *end*=None)

Computes a common time axis for several datasets. Linearly interpolate as needed. The data sets need not be the same width, and need not have the same sampling interval or a common starting time.

Parameters

data_sets: this is a list of the data to be put on a common time axis.

(*type*=*list(gpkimgclass.gpk_img)*)

start: this allows you to restrict the output data to a smaller region.

(*type*=*float* or *None*)

end: this allows you to restrict the output data to a smaller region.

(*type*=*float* or *None*)

Return Value

the *time_axis* (as a 1-D numpy array of time values), followed by a 2-D numpy array for each of the input data sets.

(*type*=*list(numpy.ndarray)* where the first *ndarray* is 1-D and the rest are two dimensional.)

commonN(*data_sets*, *start*=None, *end*=None)

Put several data sets on a common time axis. Interpolate by choosing nearest neighbor.

mul(*a*, *b*, *hdr_op*=None)**copy_interval**(*a*, *t0*, *t1*, *hdr_op*=None, *mode*="rr")

This copies the part of the time-series in *a* where $t_0 < t < t_1$. '*a*' is a *gpk_img* object.

apply(*fcn*, *a*, *hdrfcn*=lambda *x*: *x*)

Apply a function, point-by-point to the data in *a*.

```
resample(a, dt)
```

```
test()
```

116.2 Class axis

This class represents a time axis for a time series. Indices of the underlying array are assumed to be zero-based.

116.2.1 Methods

```
__init__(self, start=None, dt=None, n=None, end=None, crpix=0)
```

```
N(self)
```

```
coord(self, index)
```

What is the i^{th} coordinate if the i^{th} index==index? More plainly, this function gives you the coordinate that corresponds to the index.

```
coords(self)
```

Generate an array of all the time values.

```
index(self, t, limit=False, error=True)
```

```
indices(self, t, limit=False, error=True)
```

```
dt(self)
```

```
start(self)
```

```
end(self)
```

```
__str__(self)
```

```
__repr__(self)
```

117 Module `gmislib.wavesurfer_lab`

When used as a script, this reads label files produced by wavesurfer and prints the result.

Also contains functions that can be used for reading and writing the label format preferred by Wavesurfer.

117.1 Functions

read(*filename*, *loose*=0)

Read in label (transcription) files produced by wavesurfer.

Note that leading or trailing spaces in the label are removed.

Parameters

filename: name of label file or '-' to mean `sys.stdin`

(*type*=*str*)

loose: how many minor deviations from the ideal format are allowed

(*type*=*int*)

Return Value

(header, data). Data = [(starttime, endtime, label), ...].

write(*fd*, *header*, *data*)

Write label information to a file. Note: Expects data in [(t0, t1, label), ...] form.

Parameters

fd: where to write

(*type*=*file or file-like object*)

header: header information.

(*type*=*dict*)

data: a listing of the segments to write to the file.

(*type*=[(*segment_start_time*, *segment_end_time*, *segment_label*), ...])

117.2 Variables

Name	Description
--package--	Value: 'gmisclib'

118 Module *gmisclib.wavio*

When run as a script: `python ~/lib/wavio.py [-g gain] -wavin|-wavout infile outfile` Reads or writes .wav files from any format supported by *gpkimgclass.py* .

As a library, allows reading of class *gpk_img* data from a .WAV file, and writing class *gpk_img* to a .WAV file.

118.1 Functions

read_hdr(*fn*)

Read header information for a .WAV file. The header contains information formatted as per *gpkimgclass.gpk_img*, i.e. FITS standard information.

Parameters

fn: filename
(*type=**str*)

Return Value

dict(str: str or float or int)

read(*fn*, *tstart*=None, *t_end*=None)

Reads in a .WAV file and returns a *gpkimgclass.gpk_img* instance that contains the header and data information. If *tstart* and/or *t_end* are specified, it seeks and only reads in the necessary data between *tstart* and *t_end*.

Parameters

fn: filename
(*type=**str*)

tstart: time at which to start reading audio data
(*type=**float*)

t_end: time at which to stop reading audio data
(*type=**float*)

Return Value

audio data
(*type=**gpkimgclass instance*)

write(*data*, *fname*, *scalefac*=1, *allow_overflow*=OV_ERROR)

Parameters

- data:** is a class `gpk_img` object containing data to be written (note that the header information is ignored except for the sampling rate, bits per pixel, and number of channels).
- fname:** is the name of a file to write it to (or a file object), (*type=**str* or *file*,)
- scalefac:** is a factor to multiply the data
- allow_overflow:** can be either
- **OV_ERROR** (default, means raise a `ValueError` exception if the `data*scalefac` overflows),
 - **OV_LIMIT** (means limit the `data*scalefac` to prevent overflows – this clips the audio), or
 - **OV_IGNORE** (means let the overflows happen and don't worry.)

Raises

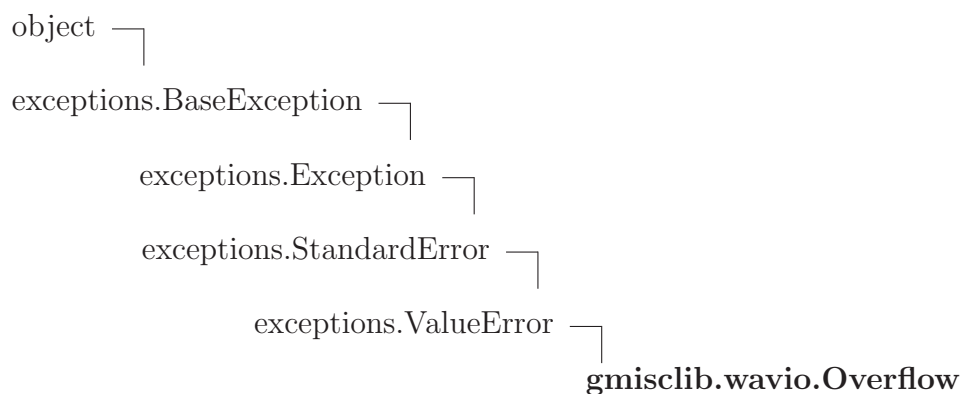
ValueError Missing information in *data*.

Overflow The output format is clipping the data. Strictly speaking, on 16-bit data, 32767 is considered clipping even though it possibly might be OK. However such extreme values are very likely to be generated by clipping at an earlier stage of the processing, so it's probably an error even if the error is not being made here.

118.2 Variables

Name	Description
OV_ERROR	Value: 0
OV_LIMIT	Value: 1
OV_IGNORE	Value: 2

118.3 Class Overflow



118.3.1 Methods

`--init--`(*self*, **s*)

x.**`--init--`**(...) initializes *x*; see *x*.**`--class--`**.**`--doc--`** for signature

Overrides: *object*.**`--init--`** *exit*(inherited documentation)

Inherited from exceptions.ValueError

`--new--`()

Inherited from exceptions.BaseException

`--delattr--`(), **`--getattr--`**(), **`--getitem--`**(), **`--getslice--`**(), **`--reduce--`**(), **`--repr--`**(),
`--setattr--`(), **`--setstate--`**(), **`--str--`**(), **`--unicode--`**()

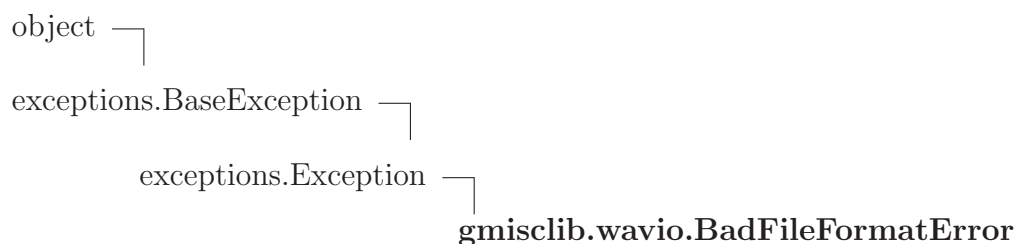
Inherited from object

`--format--`(), **`--hash--`**(), **`--reduce.ex--`**(), **`--sizeof--`**(), **`--subclasshook--`**()

118.3.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
<i>args</i> , <i>message</i>	
<i>Inherited from object</i>	
<code>--class--</code>	

118.4 Class `BadFileFormatError`



118.4.1 Methods

`__init__`(*self*, **s*)

`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `object.__init__` extit(inherited documentation)

Inherited from `exceptions.Exception`

`__new__`()

Inherited from `exceptions.BaseException`

`__delattr__`(), `__getattr__`(), `__getitem__`(), `__getslice__`(), `__reduce__`(), `__repr__`(),
`__setattr__`(), `__setstate__`(), `__str__`(), `__unicode__`()

Inherited from `object`

`__format__`(), `__hash__`(), `__reduce_ex__`(), `__sizeof__`(), `__subclasshook__`()

118.4.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
<code>args</code> , <code>message</code>	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

119 Module `gmisclib.weighted_percentile`

This computes order statistics on data with weights.

119.1 Functions

wp(*data*, *wt*, *percentiles*)

Compute weighted percentiles. If the weights are equal, this is the same as normal percentiles. Elements of the **data** and **wt** arrays correspond to each other and must have equal length (unless **wt** is **None**).

Parameters

- data:** The data.
(*type=A **numpy.ndarray** array or a **list** of numbers.*)
- wt:** How important is a given piece of data.
(*type=***None** *or a* **numpy.ndarray** *array or a* **list** *of numbers. All the weights must be non-negative and the sum must be greater than zero.*)
- percentiles:** what percentiles to use. (Not really percentiles, as the range is 0-1 rather than 0-100.)
(*type=a* **list** *of numbers between 0 and 1.*)

Return Value

the weighted percentiles of the data.
(*type=*[**float**, ... **]**)

wtd_median(*data*, *wt*)

The weighted median is the point where half the weight is above and half the weight is below. If the weights are equal, this is the same as the median. Elements of the **data** and **wt** arrays correspond to each other and must have equal length (unless **wt** is `None`).

Parameters

data: The data.

(*type=A `numpy.ndarray` array or a `list` of numbers.*)

wt: How important is a given piece of data.

(*type=`None` or a `numpy.ndarray` array or a `list` of numbers.
All the weights must be non-negative and the sum must be greater than zero.*)

Return Value

the weighted median of the data.

(*type=`float`*)

test()

test_median()

119.2 Variables

Name	Description
<code>--package--</code>	Value: <code>'gmisclib'</code>

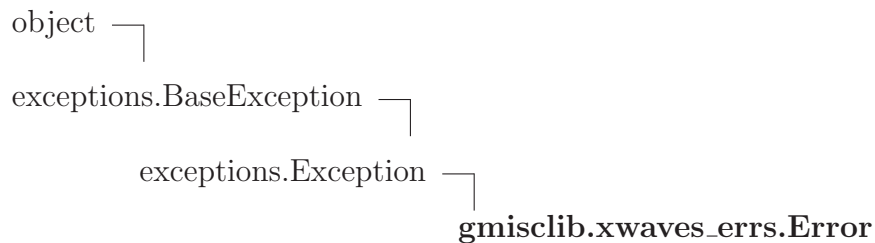
120 Module gmisclib.xwaves_errs

Errors for reading label (typically speech transcription) files.

120.1 Variables

Name	Description
<code>--package--</code>	Value: None

120.2 Class Error



Known Subclasses: gmisclib.HTK_MLF_io.ReferencedFileNotFound, gmisclib.xwaves_errs.BadFileFormat, gmisclib.xwaves_errs.DataError, gmisclib.xwaves_errs.NoSuchFileError

120.2.1 Methods

`--init--(self, *x)`

`x.--init--(...)` initializes `x`; see `x.--class--.--doc--` for signature

Overrides: `object.--init--` extit(inherited documentation)

Inherited from exceptions.Exception

`--new--()`

Inherited from exceptions.BaseException

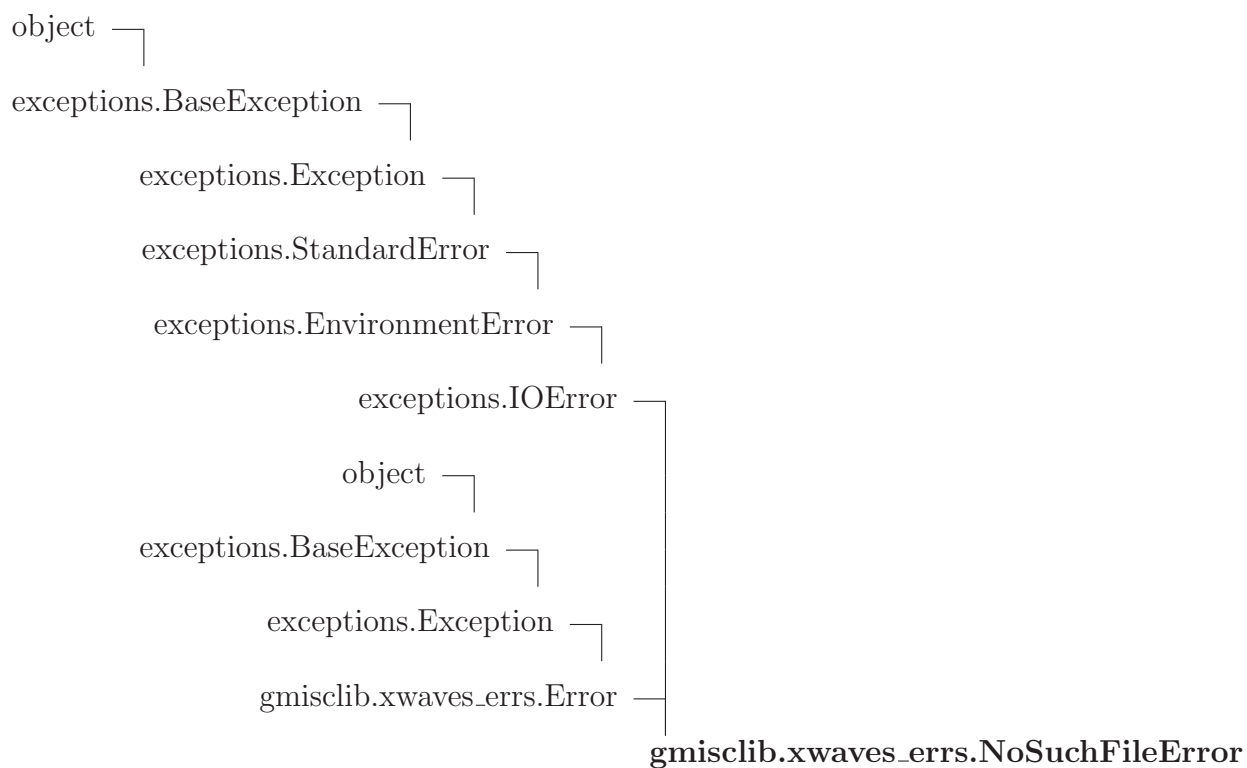
`--delattr--()`, `--getattribute--()`, `--getitem--()`, `--getslice--()`, `--reduce--()`, `--repr--()`,
`--setattr--()`, `--setstate--()`, `--str--()`, `--unicode--()`

Inherited from object

`--format--()`, `--hash--()`, `--reduce_ex--()`, `--sizeof--()`, `--subclasshook--()`

120.2.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
<code>args</code> , <code>message</code>	
<i>Inherited from object</i>	
<code>__class__</code>	

120.3 Class `NoSuchFileError`**120.3.1 Methods**

```

__init__(self, *x)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature
Overrides: object.__init__ exitit(inherited documentation)

```

Inherited from `exceptions.IOError`

```
__new__()
```

Inherited from `exceptions.EnvironmentError`

`__reduce__()`, `__str__()`

Inherited from `exceptions.BaseException`

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__repr__()`, `__setattr__()`,
`__setstate__()`, `__unicode__()`

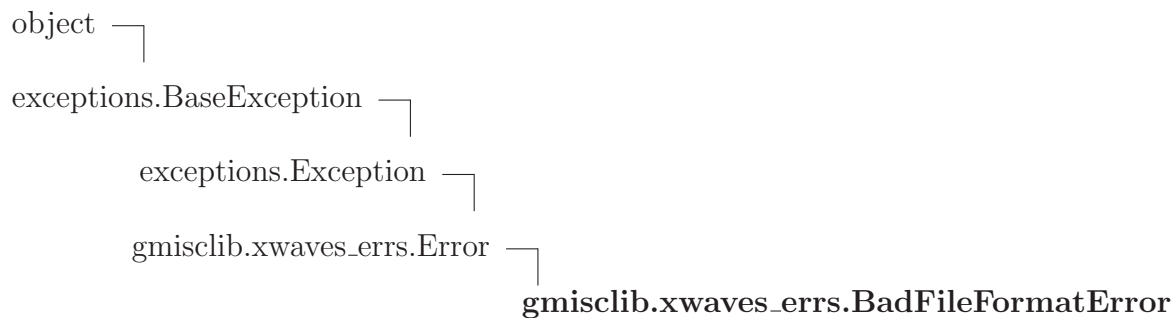
Inherited from `object`

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

120.3.2 Properties

Name	Description
<i>Inherited from <code>exceptions.EnvironmentError</code></i>	<code>errno</code> , <code>filename</code> , <code>strerror</code>
<i>Inherited from <code>exceptions.BaseException</code></i>	<code>args</code> , <code>message</code>
<i>Inherited from <code>object</code></i>	<code>__class__</code>

120.4 Class `BadFileFormatError`



120.4.1 Methods

`__init__(self, *x)`
`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature
 Overrides: `object.__init__` `exitit`(inherited documentation)

Inherited from `exceptions.Exception`

`__new__()`

Inherited from `exceptions.BaseException`

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

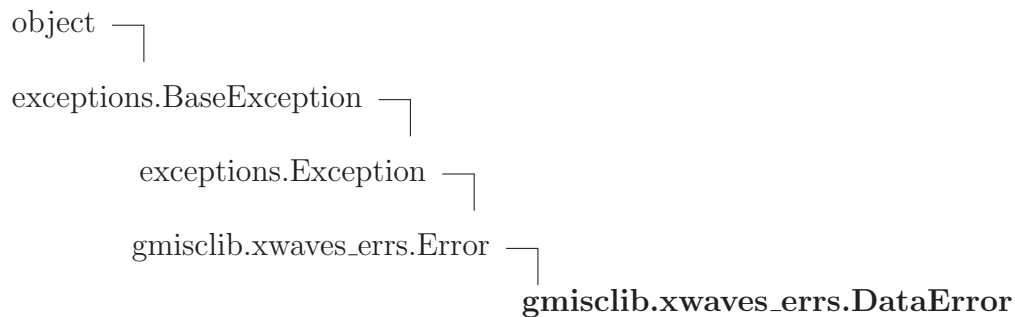
Inherited from `object`

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

120.4.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
<code>args</code> , <code>message</code>	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

120.5 Class `DataError`



Known Subclasses: `gmisclib.xwaves.errs.DataOutOfOrderError`

120.5.1 Methods

`__init__`(*self*, **x*)
x.`__init__`(...) initializes *x*; see *x*.`__class__`.`__doc__` for signature
 Overrides: `object.__init__` extit(inherited documentation)

Inherited from `exceptions.Exception`

`__new__()`

Inherited from `exceptions.BaseException`

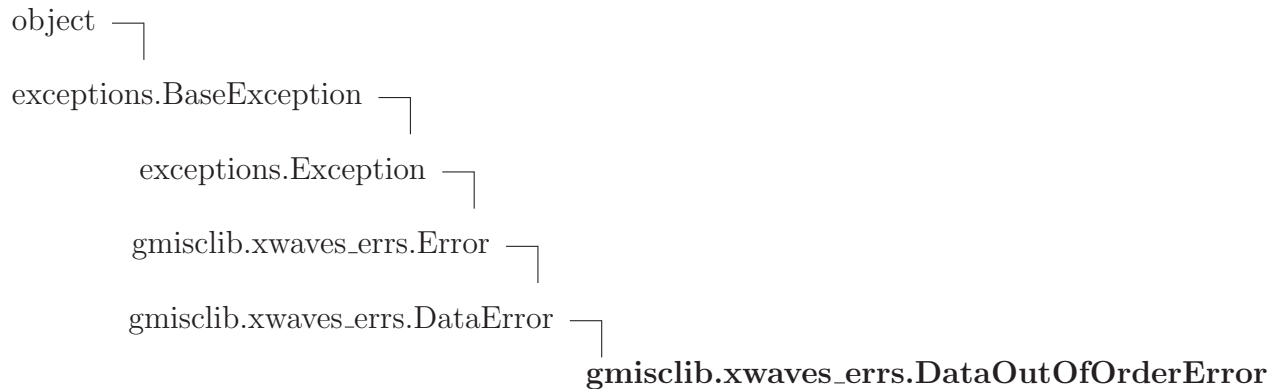
`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

Inherited from `object`

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

120.5.2 Properties

Name	Description
<i>Inherited from <code>exceptions.BaseException</code></i>	
<code>args</code> , <code>message</code>	
<i>Inherited from <code>object</code></i>	
<code>__class__</code>	

120.6 Class `DataOutOfOrderError`**120.6.1 Methods**

`__init__(self, *s)`
`x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signature
 Overrides: `object.__init__` `exitit`(inherited documentation)

Inherited from `exceptions.Exception`

`__new__()`

Inherited from `exceptions.BaseException`

`__delattr__()`, `__getattr__()`, `__getitem__()`, `__getslice__()`, `__reduce__()`, `__repr__()`,
`__setattr__()`, `__setstate__()`, `__str__()`, `__unicode__()`

Inherited from object

`__format__()`, `__hash__()`, `__reduce_ex__()`, `__sizeof__()`, `__subclasshook__()`

120.6.2 Properties

Name	Description
<i>Inherited from exceptions.BaseException</i>	
args, message	
<i>Inherited from object</i>	
<code>__class__</code>	

121 Module *gmisclib.xwaves_lab*

Reads label files produced by ESPS xwaves. Writes labels files that ESPS xwaves can read.

121.1 Functions

read (<i>filename</i> , <i>loose</i> =0)
--

read in .lab files produced by ESPS xlabel. returns (header, data). Data = [(time, label), ...]. Note that leading or trailing spaces in the label are removed.

start_stop (<i>d</i> , <i>dropfirst</i> =False)

Converts data in (end_time, label) format to (t_start, t_stop, label)

end_marks (<i>d</i> , <i>beglabel</i> , <i>delta</i> =0.001, <i>interlabel</i> =None)

Converts data in a (start, stop, label) format to (end_time, label). It introduces extra labels if neccessary. A beginning label is required, to mark the start of the first segment.

write (<i>fd</i> , <i>header</i> , <i>data</i>)
--

Expects data in [(t, label), ...] form.

121.2 Variables

Name	Description
<code>__package__</code>	Value: 'gmisclib'

122 Module *gmisclib.xwaves_mark*

Read in .in files produced by ESPS xmark. Returns (header, data), where data is (time, word_or_phoneme, type), where type is 0 for words, 1 for phonemes. Times are guaranteed to be increasing inside the set of all words, and also inside the set of all phonemes. Word marks precede the corresponding phoneme marks.

122.1 Functions

write(*fd, hdr, data, SortData=1*)

read(*filename*)

Read in .in files produced by ESPS xmark. Returns (header, data), where data is (time, word_or_phoneme, type), where type is 0 for words, 1 for phonemes. Times are guaranteed to be increasing inside the set of all words, and also inside the set of all phonemes. Word marks precede the corresponding phoneme marks.

mark_to_lab(*data, ty*)

This function lets you take a mixed list of word and phone labels, such as provided by *xwaves_mark.read()*, and will select out one or the other type. Type is PHONE or WORD.

combine_2labs(*whdr, wd, phdr, pd, utterance, TOL=0.001*)

Combine two XLAB files into one XMARK file. It forces (within rounding errors) the words to enclose the phones.

122.2 Variables

Name	Description
DecrTol	Value: 0.004999
PHONE	Value: 1
WORD	Value: 0
__doc__	Value: read.__doc__
__package__	Value: 'gmisclib'

123 Module gpk_wavio

When run as a script: `python ~/lib/wavio.py [-g gain] -wavin|-wavout infile outfile` Reads or writes .wav files from any format supported by `gpkimgclass.py` .

As a library, allows reading of class `gpk_img` data from a .WAV file, and writing class `gpk_img` to a .WAV file.

124 Module *mcmc*

Adaptive Markov-Chain Monte-Carlo algorithm. This can be used to generate samples from a probability distribution, or also as a simulated annealing algorithm for maximization. This can be used as a script, from the command line, or (more likely) it can be imported and its functions and classes can be used.

The central interfaces are the `BootStepper` class, and within it, the `step()` method is used iteratively to take a Markov step.

To use it as a script, you create your own module that implements a few functions, and run `mcmc.py`. It will import your module and call your functions repeatedly.

The algorithm is described in Kochanski and Rosner 2010, and earlier versions have been used in ZZZ. It evolved originally from `amoeba_anneal` (in Numerical Recipes, Press et al.). The essential feature is that it keeps a large archive of previous positions (possibly many times more than N of them). It samples two positions from the archive and subtracts them to generate candidate steps. This has the nice property that when sampling from a multivariate Gaussian distribution, the candidate steps match the distribution nicely.

It can be operated in two modes (or set to automatically switch). One is optimization mode where it heads for the maximum of the probability distribution. The other mode is sampling mode, where it asymptotically follows a Markov sampling procedure and has the proper statistical properties.

As a script, it takes the following arguments:

- `-o fff` Specifies a log file for some basic logging.
- `{C -py ddd/mm}` Specifies the module to load and to optimize. If the name contains a slash, it searches `ddd` instead of the python search path, looking for a module named `mm`. If no slash, then it looks up the module in `PYTHONPATH`.
- `-NI NN` Sets the number of iterations. Required, unless the module defines `yourmod.NI`.
- `--` Stops interpretation of the argument list. The remainder is passed to `yourmod.init()`, if it is defined, then to `yourmod.start()`, if that is defined.

It uses the following functions and variables:

- `yourmod.c` (required). This is an object that is passed into `yourmod.resid()` or `yourmod.logp()`. You can make it be anything you want, as it is only touched by your functions. Often, it contains data or parameters for these functions, but it can be `None` if you have no need for it.
- `yourmod.start(arglist, c)` (Not required). This is where you initialize the Monte-Carlo iteration. If you define `yourmod.start()`, it is passed the `arglist`, after the standard `mcmc.py` flags and arguments are removed from it. `Yourmod.start()` must return a list of one or more starting vectors, (either a list of numpy arrays or a list of sequences of floats). These starting vectors are used to seed the iteration.

If it is not defined, `mcmc.py` will read a list of starting vectors in ascii format from the standard input using `_read()`.

- `yourmod.init(ndim, ni, c, arglist)` (Not required). This is primarily there to open a log file. `Init` is passed the remainder of the scripts argument list, after `yourmod.start()` looks at it. It can do anything it wishes with the argument list. It can return an object or `None`.

If it returns something other than `None`, it will be used this way:

```
x = yourmod.init(ndim, ni, yourmod.c, arglist) x.add(parameters, iteration) # add is
there to accumulate averages of parameters # or to log parameter vectors. x.finish(sys.stdout)
# finish can be used to print a summary or close a log file.
```

`Ndim` is the length of the parameter vector, `ni` is the number of iterations, `c` is `yourmod.c`.

- `yourmod.resid(x, c)` (Either `yourmod.logp()` or `yourmod.resid()` is required.)
- `yourmod.logp(x, c)` (Either `yourmod.logp` or `yourmod.resid` is required.) These functions are the thing which is to be optimized. `Yourmod.log` returns the logarithm (base e) of the probability density at position `x`. (It does not need to be normalized, so it really only needs something proportional to the probability density.) `X` is a numpy array (a 1-dimensional vector), and `c` is an arbitrary object that you define.

`Yourmod.logp` should return `None` if `x` is a silly value. Either that, or it can raise a `NotGoodPosition` exception. Note that the optimizer will tend to increase values of `logp`. It's a maximizer, not a minimizer!

If you give `yourmod.resid()` instead, `logp()` will be calculated as -0.5 times the sum of the squares of the residuals. It should return a numpy vector. It may return `None`, or raise `NotGoodPosition`, which will cause the optimizer to treat the position as extremely bad.

- `yourmod.log(result_of_init, i, p, w)` (Not required). This is basically a function to log your data. No return value. It is called at every step and passed the result of `yourmod.init()` (which might be a file descriptor to write into). It is also passed the iteration count, `i`, the current parameter vector, `p`, and `w`, which is reserved for possible future use.
- `yourmod.NI` (required unless the `-NI` argument is given on the command line) Specifies how many iterations. This is an integer.
- `yourmod.STARTLOW` (not required). Integer 0 or 1. If true, the iteration is started from the largest value of `logp` that is known.
- `yourmod.V` (Not required) This specifies the covariance matrix that is used to hop from one test point to another. It is a function that takes the result of `yourmod.start()` and returns a Numeric/numarray 2-dimensional square matrix,

If it is not specified, `mcmc` will use some crude approximation and will probably start more slowly, but should work OK. If it is not specified and `yourmod.start()` returns only one starting position, then it will just use an identity matrix as the covariance matrix, and it might still work, but don't expect too much.

124.1 Functions

`logp_from_resid(x, c, resid_fcn)`

`go(argv, theStepper)`

Run the optimization, controlled by command line flags.

124.2 Variables

Name	Description
Debug	Value: 0
MEMORYS_WORTH_OF_PARAMETERS	Value: 1000000000.0
<code>__package__</code>	Value: None

124.3 Class `def_logger`

124.3.1 Methods

`__init__(self, ndim, ni, c, arglist)`

`add(self, p, i)`

`finish(self, stdout)`

125 Module q3html

This program takes files named *.q in the current directory, and converts them to HTML. To do that, it looks for a python script called 'headfoot.py' or '../headfoot.py' or This script needs to define one variable (server) and two functions (header and footer).

The .q files have the following format:

```
TITLE title string
OTHER_HEADER_KEYWORD header info

P
    Text is 3>5 ?
    A ref="foo.html" No!
        More text inside the link.
    IMG ref="foo.gif"
    UL
        LI List item
        LI Another List item
```

The header info is separated from the body with a blank line. HTML tags are always the first thing on a line. Line continuations and enclosures are indicated with indentation. Tags automatically close when the indentation gets smaller.

The PRE tag is the only exception. You have to close it with a line beginning '/PRE'. No indenting is needed inside a PRE /PRE pair.

The user needs to define one thing: the address of the web server. This address is prepended to all hyperlinks and image references that aren't absolute.

All the header information gets put into a dictionary, and is passed into the header() and footer() functions. That dictionary has 'filename' and '_server_root' entries added.

The user can also define three things:

DEFAULT_HEADER, a dictionary passed to the header() and footer functions(). Useful things to put in DEFAULT_HEADER are:

- 'lang': 'en' # appears in the <html lang="en"> tag
- 'stylesheet': url becomes a link to a stylesheet: <link rel="stylesheet" type="text/css" href="url">
- Anything in the form M.x:y becomes a <meta name="x" content="y">
- M.description and M.keywords are useful.
- Anything in the form HTTP.x:y becomes a <meta http-equiv="x" content="y">

Header(thd, hinfo) takes two arguments, a file descriptor to which it should write the HTML, and a dictionary of header information.

Footer(thd, hinfo) takes the same arguments.
See _header() and _footer() below, for examples.

125.1 Functions

qfiles(*dir*)

av(*k*, *v*)

measure_indent(*s*)

prepare_text(*l*)

starts_with_a_tag(*l*)

tokenize(*l*)

format_dict(*d*, *drop*=None)

dequote(*a*)

urlq(*s*)

prepend(*a*, *b*)

Process a URL. Pathnames are prepended with the server root. Complete URLs are untouched.

process_tag(*d*, *od*, *hinfo*, *eol*)

escape(*s*)

get_logical_line(*lines*)

process(*lines*, *od*, *hinfo*)

swapend(*a*, *e*)

aget(*dic*, *alist*)

```
sw(s, prefix)
```

```
dpre(s, prefix)
```

```
agp(dic, prefix)
```

```
get(dic, key, df)
```

```
readheader(fd, t)
```

```
needcopy(a, b, force=None)
```

Do we need to reconstruct file b from file a? Also reconstruct b if it is older than the date in force.

```
file_only(url)
```

Given a URL, just return the file part, not the target specifier inside the file.

```
del_fin_sl(s)
```

```
go(env, force=None)
```

```
easygo()
```

125.2 Variables

Name	Description
XML	Value: 0
POINTTAG	Value: 1
LL	Value: 60
TABW	Value: 8
DEFAULT_TAG	Value: 'P'
DEFAULT_INDENT	Value: -1
MTIME	Value: 8
DEFF	Value: 'headfoot.py'
__package__	Value: None

125.3 Class *lineC*

125.3.1 Methods

<code>__init__(self, l)</code>

<code>getarg(self, k, defv)</code>

126 Module *run_several*

This script runs a bunch of processes in parallel. You tell it what to run on the standard input. It spawns off a specified number of subprocesses, and hands the jobs to the subprocesses.

Usage: *run_several.py* [-v] Ncpu Oom_adjust <list_of_commands> >log_of_outputs

Flags: -v means "be more verbose".

Ncpu is the maximum number of cores to devote to running these tasks (you can enter zero to have it use all the cores of the local machine), or +N or -N to have it use N more (or less) cores than the machine has. Further x and * allow you to multiply the number of CPUs by a factor. (NB: The +, -, x, * forms always leave at least one core running, no matter what N is or how few cores you have.) All this can be useful if you want to avoid loading the machine too heavily, or if you want to boost the loading because your jobs are limited by disk seek time.

Oom_adjust is a number given to Linux's OOM killer. When the system starts running out of memory, the OOM killer kills processes. Positive numbers (e.g. 3) make it more likely that a process will be killed. So, if your subtasks are likely to use too much memory and are not critical, set Oom_adjust positive.

The commands in list_of_commands are just piped into Bash's standard input one at a time. Each line is (typically) processed by a different instance of bash, so don't try any multi-line commands. The resulting standard outputs and standard errors are kept separate and each processes' outputs are printed as a lump when it completes. Stderr from a subprocess comes out as stdout from *run_several.py*; various blocks of output are separated by lines with hash marks and strings of "—" (see *write_msgs()*).

Note that the processes may finish in any arbitrary order. The integer on the stdout and stderr separator lines gives you the (zero-based) line number in the input file. *Run_several.py* returns failure if any of its subprocesses fail, and it will terminate on the first failure.

Example 1:

```
$ echo "pwd" | run_several.py 0 0
# command 0: pwd
# stdout 0 -----(exited with 0)
/home/gpk/speechresearch/gmisc/lib/bin
# stderr 0 -----
$
```

Example 2:

```
$ { echo uname -m; echo uname -p; echo uname -o; } | run_several.py 0 0
# command 0: uname -m
# stdout 0 -----(exited with 0)
```

```

x86_64
# stderr 0 -----
# command 1: uname -p
# stdout 1 -----(exited with 0)
unknown
# stderr 1 -----
# command 2: uname -o
# stdout 2 -----(exited with 0)
GNU/Linux
# stderr 2 -----
$

```

126.1 Functions

write_msgs(*p*, *exitstatus*)

This produces the output.

get_loadavg()

mem_pressure(*np*)

Return Value

True if the system is running low on memory.

(*type=bool*)

get_ncpu()

wait_until_unloaded()

Wait a while until the system becomes less loaded. It won't wait forever, though.

set_oom(*pid*, *ooma*)

run_processes(*fd*, *np*, *ooma*, *verbose*)

parse_NP(*s*)

126.2 Variables

Name	Description
DT	Value: 1.0
--package--	Value: None

127 Module *select_fiat_entries*

This program reads a FIAT format data file and displays some of the information. You can select some of the lines with the `-select` flag, and the display the selected lines in several different ways.

Flags:

- `-select expr` Where `expr` is a python expression that returns True or False. All the normal builtins are available. The data in the line is presented as a set of local variables, so that if the value in the column labelled "filename" is "f005.wav", then you could test for `'filename=="f005.wav"'`. Note that everything is a string, so you may need to convert to int or float before some tests: `'int(nspeakers)<5'`.
- `-showav` Displays the entire line in `a=v; format`.
- `-showfiat` Displays the resulting file in FIAT format, preserving the header information.
- `-show format` This displays the file as a text file, using evaluating format (which is a snippet of python code that should return a string): `-show "%s # %s'%(fname,loudness)'` will display the `fname` and `loudness` columns of the selected lines, separated by a hash mark.
- `-rav` Reads the input file in `a=v; format` rather than `fiat`.
- `-senv` Exec some python code to set a global variable that is visible in the `-select` expression. (e.g. `"x=3"`).
- `-penv` Exec some python code to set a global variable that is visible in the `-show` expression. (e.g. `"x=3"`).
- `-strap` exception value

127.1 Functions

```
process(f, read_fcn, o, select)
```

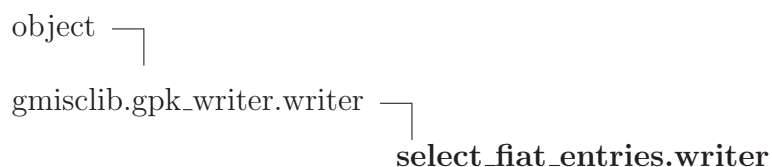
```
fiat_read(fn)
```

```
avio_read(fn)
```

127.2 Variables

Name	Description
SHOWAV	Value: 1
SHOWFIAT	Value: 2
__package__	Value: None

127.3 Class writer



127.3.1 Methods

comment(*self*, *comment*)

Add a comment to the data file.

Overrides: gmisclib.gpk_writer.writer.comment

header(*self*, *k*, *v*)

Overrides: gmisclib.gpk_writer.writer.header

__init__(*self*, *fd*, *s*)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Overrides: object.__init__ extit(inherited documentation)

datum(*self*, *data_item*)

Overrides: gmisclib.gpk_writer.writer.datum

globals(*self*, *code*)

append(*self*, *d*)

close(*self*)

comments(*self*, *comments*)

Add comments to the data file. Comments can appear anywhere.

data(*self*, *dataset*)

Write a series of lines to the output file.

extend(*self*, *d*)

<code>flush(<i>self</i>)</code>

<code>headers(<i>self</i>, <i>h</i>)</code>

Inherited from object

`--delattr--()`, `--format--()`, `--getattr--()`, `--hash--()`, `--new--()`, `--reduce--()`, `--reduce_ex--()`,
`--repr--()`, `--setattr--()`, `--sizeof--()`, `--str--()`, `--subclasshook--()`

127.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>--class--</code>	

Index

ds9, 166

gmisclib (*package*), 24–28

gmisclib.accent_spec (*module*), 61–63

gmisclib.accent_spec.BadMatchError (*class*),
62–63

gmisclib.accent_spec.prefix (*function*), 61

gmisclib.accent_spec.preshow (*function*),
61

gmisclib.accent_spec.suffix (*function*), 61

gmisclib.accent_spec.sufshow (*function*),
62

gmisclib.accent_spec.test (*function*), 62

gmisclib.array_window (*module*), 64–65

gmisclib.array_window.array_window (*class*),
64–65

gmisclib.array_window.test (*function*), 64

gmisclib.avio (*module*), 66–69

gmisclib.avio.BadFormatError (*class*), 67–
68

gmisclib.avio.concoct (*function*), 66

gmisclib.avio.parse (*function*), 66

gmisclib.avio.read (*function*), 66

gmisclib.avio.read_hdc (*function*), 67

gmisclib.avio.test1 (*function*), 66

gmisclib.avio.test2 (*function*), 67

gmisclib.avio.writer (*class*), 68–69

gmisclib.bark_scale (*module*), 70

gmisclib.bark_scale.acosh (*function*), 70

gmisclib.bark_scale.asinh (*function*), 70

gmisclib.bark_scale.bark_to_f (*function*),
70

gmisclib.bark_scale.cbw (*function*), 70

gmisclib.bark_scale.f_to_bark (*function*),
70

gmisclib.beamsearch (*module*), 71

gmisclib.beamsearch.beamsearch (*func-*
tion), 71

gmisclib.blue_data_attributes (*module*), 72–
73

gmisclib.blue_data_attributes.blue_attributes
(*class*), 72–73

gmisclib.blue_data_attributes.test (*func-*
tion), 72

gmisclib.blue_data_selector (*module*), 74–
77

gmisclib.blue_data_selector.bluedata (*class*),
74–77

gmisclib.blue_data_selector.test (*function*),
74

gmisclib.cache (*module*), 78

gmisclib.cache.cachepath (*function*), 78

gmisclib.chunkio (*module*), 79–89

gmisclib.chunkio.BadFileFormat (*class*),
79–80

gmisclib.chunkio.chunk (*class*), 80–81

gmisclib.chunkio.chunk_w (*class*), 85–86

gmisclib.chunkio.chunkstring_w (*class*),
87–89

gmisclib.chunkio.datachunk (*class*), 81–
83

gmisclib.chunkio.datachunk_w (*class*), 86–
87

gmisclib.chunkio.stringchunk (*class*), 83–
85

gmisclib.chunkio.test (*function*), 79

gmisclib.chunkio.test_e (*function*), 79

gmisclib.convex_hull2d (*module*), 90–91

gmisclib.convex_hull2d.convexHull (*func-*
tion), 90

gmisclib.convex_hull2d.DuplicatePoints
(*class*), 90–91

gmisclib.convex_hull2d.saveAsEps (*func-*
tion), 90

gmisclib.convex_hull2d.test (*function*), 90

gmisclib.dict_vector (*module*), 92–95

gmisclib.dict_vector.dict_vector (*class*),
92–95

gmisclib.dict_vector.log (*function*), 92

gmisclib.dict_vector.max_between (*func-*
tion), 92

gmisclib.dict_vector.max_within (*function*),
92

gmisclib.dict_vector.min_between (*func-*

- tion*), 92
- gmisclib.dict_vector.min_within (*function*), 92
- gmisclib.dict_vector.round (*function*), 92
- gmisclib.dictops (*module*), 96–107
 - gmisclib.dictops.add_doc (*function*), 98
 - gmisclib.dictops.add_dol (*function*), 98
 - gmisclib.dictops.argmax (*function*), 97
 - gmisclib.dictops.BadFileFormatError (*class*), 98–99
 - gmisclib.dictops.compose (*function*), 97
 - gmisclib.dictops.dict_of_accums (*class*), 102–103
 - gmisclib.dictops.dict_of_averages (*class*), 103–104
 - gmisclib.dictops.dict_of_lists (*class*), 99–100
 - gmisclib.dictops.dict_of_maxes (*class*), 104–105
 - gmisclib.dictops.dict_of_sets (*class*), 100–102
 - gmisclib.dictops.dict_of_X (*class*), 105–106
 - gmisclib.dictops.filter (*function*), 96
 - gmisclib.dictops.list_of_dicts (*class*), 106–107
 - gmisclib.dictops.read2c (*function*), 97
 - gmisclib.dictops.remove (*function*), 96
 - gmisclib.dictops.rev1to1 (*function*), 96
- gmisclib.die (*module*), 108–110
 - gmisclib.die.catch (*function*), 108
 - gmisclib.die.catchexit (*function*), 108
 - gmisclib.die.counter (*class*), 110
 - gmisclib.die.dbg (*function*), 108
 - gmisclib.die.die (*function*), 108
 - gmisclib.die.exit (*function*), 109
 - gmisclib.die.get (*function*), 109
 - gmisclib.die.info (*function*), 108
 - gmisclib.die.note (*function*), 109
 - gmisclib.die.warn (*function*), 108
- gmisclib.die2 (*module*), 111–112
 - gmisclib.die2.catch (*function*), 111
 - gmisclib.die2.catchexit (*function*), 111
 - gmisclib.die2.dbg (*function*), 111
 - gmisclib.die2.die (*function*), 111
 - gmisclib.die2.exit (*function*), 111
 - gmisclib.die2.get (*function*), 111
 - gmisclib.die2.info (*function*), 111
 - gmisclib.die2.note (*function*), 111
 - gmisclib.die2.show (*function*), 111
 - gmisclib.die2.warn (*function*), 111
- gmisclib.do_exec (*module*), 113
 - gmisclib.do_exec.get (*function*), 113
 - gmisclib.do_exec.getall (*function*), 113
- gmisclib.do_pgplot (*module*), 114
 - gmisclib.do_pgplot.plot (*class*), 114
- gmisclib.ds9_region (*module*), 115
 - gmisclib.ds9_region.writer (*class*), 115
- gmisclib.dyn_prog (*module*), 116
 - gmisclib.dyn_prog.path (*function*), 116
 - gmisclib.dyn_prog.test1 (*function*), 116
 - gmisclib.dyn_prog.test2 (*function*), 116
 - gmisclib.dyn_prog.test3 (*function*), 116
- gmisclib.edit_distance (*module*), 117–121
 - gmisclib.edit_distance.def_cost (*function*), 118
 - gmisclib.edit_distance.dist (*function*), 117
 - gmisclib.edit_distance.distf1 (*function*), 117
 - gmisclib.edit_distance.distf2 (*function*), 117, 118
 - gmisclib.edit_distance.free_del (*function*), 119
 - gmisclib.edit_distance.free_sub (*function*), 118
 - gmisclib.edit_distance.test (*function*), 119
 - gmisclib.edit_distance.test2 (*function*), 119
 - gmisclib.edit_distance.text_cost (*class*), 119–121
- gmisclib.entropy (*module*), 122
 - gmisclib.entropy.information_gained (*function*), 122
 - gmisclib.entropy.multinomial_fixer (*function*), 122
 - gmisclib.entropy.multinomial_logp (*function*), 122
 - gmisclib.entropy.P (*function*), 122

- gmisclib.erb_scale (*module*), 123
 - gmisclib.erb_scale.ebw (*function*), 123
 - gmisclib.erb_scale.erb_to_f (*function*), 123
 - gmisclib.erb_scale.f_to_erb (*function*), 123
- gmisclib.evolution (*module*), 124
- gmisclib.fake_file (*module*), 125
 - gmisclib.fake_file.file (*class*), 125
 - gmisclib.fake_file.open (*function*), 125
- gmisclib.fiat_merge (*module*), 126
 - gmisclib.fiat_merge.merge (*function*), 126
- gmisclib.fiatio (*module*), 127–139
 - gmisclib.fiatio.col_order (*function*), 128
 - gmisclib.fiatio.ConflictingColumnSpecification (*class*), 138–139
 - gmisclib.fiatio.FiatioWarning (*class*), 133–134
 - gmisclib.fiatio.merged_writer (*class*), 136–138
 - gmisclib.fiatio.read (*function*), 129
 - gmisclib.fiatio.read_as_float_array (*function*), 132
 - gmisclib.fiatio.read_merged (*function*), 130
 - gmisclib.fiatio.readiter (*function*), 131
 - gmisclib.fiatio.shared_data_values (*function*), 129
 - gmisclib.fiatio.test (*function*), 133
 - gmisclib.fiatio.test1 (*function*), 132
 - gmisclib.fiatio.test2 (*function*), 133
 - gmisclib.fiatio.test3 (*function*), 133
 - gmisclib.fiatio.test4 (*function*), 133
 - gmisclib.fiatio.write (*function*), 129
 - gmisclib.fiatio.write_array (*function*), 128
 - gmisclib.fiatio.writer (*class*), 134–136
- gmisclib.find_home (*module*), 140
 - gmisclib.find_home.data (*function*), 140
 - gmisclib.find_home.directory (*function*), 140
 - gmisclib.find_home.executable (*function*), 140
 - gmisclib.find_home.module (*function*), 140
 - gmisclib.find_home.os_prgm (*function*), 140
- gmisclib.find_ngram (*module*), 141–142
 - gmisclib.find_ngram.find (*function*), 141
 - gmisclib.find_ngram.find_lab (*function*), 141
 - gmisclib.find_ngram.find_mark (*function*), 141
 - gmisclib.find_ngram.matches (*function*), 141
- gmisclib.findleak (*module*), 143
 - gmisclib.findleak.alloc1k (*function*), 143
 - gmisclib.findleak.get_refcounts (*function*), 143
 - gmisclib.findleak.lookvmstat (*function*), 143
 - gmisclib.findleak.print_top_N (*function*), 143
 - gmisclib.findleak.readvm (*function*), 143
 - gmisclib.findleak.vm (*function*), 143
- gmisclib.ftest (*module*), 144–145
 - gmisclib.ftest.agammln (*function*), 144
 - gmisclib.ftest.betacf (*function*), 144
 - gmisclib.ftest.betai (*function*), 144
 - gmisclib.ftest.fprob (*function*), 144
 - gmisclib.ftest.gammln (*function*), 144
- gmisclib.fuzzygraph (*module*), 146
 - gmisclib.fuzzygraph.addcurve (*function*), 146
- gmisclib.g2_select (*module*), 147–149
 - gmisclib.g2_select.accept (*function*), 147
 - gmisclib.g2_select.evaluate (*function*), 147
 - gmisclib.g2_select.filter_iter (*function*), 147
 - gmisclib.g2_select.filterlist (*function*), 147
 - gmisclib.g2_select.selector_c (*class*), 148–149
 - gmisclib.g2_select.test (*function*), 148
 - gmisclib.g2_select.why (*function*), 147
 - gmisclib.g2_select.whynot (*function*), 147
- gmisclib.g_closure (*module*), 150–152
 - gmisclib.g_closure.ArgUnspecifiedError (*class*), 150–151
 - gmisclib.g_closure.Closure (*class*), 151–152
 - gmisclib.g_closure.NotYet (*class*), 150
- gmisclib.g_datetime (*module*), 153
 - gmisclib.g_datetime.ISO2datetime (*function*), 153

- gmisclib.g_datetime.timedelta2float (*function*), 153
- gmisclib.g_ds9 (*module*), 154–168
 - gmisclib.g_ds9.BadTagError (*class*), 158–159
 - gmisclib.g_ds9.circle (*class*), 161–162
 - gmisclib.g_ds9.copy (*function*), 154
 - gmisclib.g_ds9.dequote (*function*), 154
 - gmisclib.g_ds9.ds9 (*class*), 165–167
 - gmisclib.g_ds9.ds9_file (*class*), 167–168
 - gmisclib.g_ds9.ds9_io (*class*), 157–158
 - gmisclib.g_ds9.execwait (*class*), 156–157
 - gmisclib.g_ds9.line (*class*), 160–161
 - gmisclib.g_ds9.make_chunks (*function*), 154
 - gmisclib.g_ds9.point (*class*), 162–163
 - gmisclib.g_ds9.property_parser (*function*), 154
 - gmisclib.g_ds9.r_list_factory (*function*), 154
 - gmisclib.g_ds9.read_regions (*function*), 154
 - gmisclib.g_ds9.read_regions_iter (*function*), 154
 - gmisclib.g_ds9.Region (*class*), 159–160
 - gmisclib.g_ds9.region_parser (*function*), 154
 - gmisclib.g_ds9.test_io (*function*), 154
 - gmisclib.g_ds9.test_plot (*function*), 155
 - gmisclib.g_ds9.test_property_parser (*function*), 154
 - gmisclib.g_ds9.test_region_parser (*function*), 154
 - gmisclib.g_ds9.test_tags (*function*), 155
 - gmisclib.g_ds9.test_text_kluge (*function*), 154
 - gmisclib.g_ds9.text (*class*), 163–165
 - gmisclib.g_ds9.text_kluge (*function*), 154
 - gmisclib.g_ds9.UnknownProperty (*class*), 155–156
- gmisclib.g_encode (*module*), 169–170
 - gmisclib.g_encode.BadFormatError (*class*), 169–170
 - gmisclib.g_encode.encoder (*class*), 170
 - gmisclib.g_encode.test (*function*), 169
- gmisclib.g_entropy (*module*), 171
 - gmisclib.g_entropy.entropy_probs (*function*), 171
 - gmisclib.g_entropy.entropy_vec (*function*), 171
- gmisclib.g_exec (*module*), 172–177
 - gmisclib.g_exec.ExecError (*class*), 176–177
 - gmisclib.g_exec.get (*function*), 174
 - gmisclib.g_exec.get_raw (*function*), 175
 - gmisclib.g_exec.getall (*function*), 175
 - gmisclib.g_exec.getiter (*function*), 173
 - gmisclib.g_exec.getiter_raw (*function*), 173
 - gmisclib.g_exec.getlast (*function*), 175
 - gmisclib.g_exec.test (*function*), 176
- gmisclib.g_implements (*module*), 178–181
 - gmisclib.g_implements.check (*function*), 179
 - gmisclib.g_implements.GITTypeError (*class*), 180–181
 - gmisclib.g_implements.impl (*function*), 178
 - gmisclib.g_implements.make_optional (*function*), 179
 - gmisclib.g_implements.make_strict (*function*), 179
 - gmisclib.g_implements.make_varargs (*function*), 179
 - gmisclib.g_implements.make_vartype (*function*), 179
 - gmisclib.g_implements.Strict (*function*), 178
 - gmisclib.g_implements.test (*function*), 179
 - gmisclib.g_implements.Vartype (*function*), 178
 - gmisclib.g_implements.why (*function*), 178
- gmisclib.g_keyed_accum (*module*), 182
 - gmisclib.g_keyed_accum.keyed_avg (*class*), 182
 - gmisclib.g_keyed_accum.test (*function*), 182
- gmisclib.g_lin_fit (*module*), 183–184
 - gmisclib.g_lin_fit.fit (*function*), 183
 - gmisclib.g_lin_fit.test (*function*), 183

- gmisclib.g_lin_fit.test1 (*function*), 183
- gmisclib.g_lin_fit.test2 (*function*), 183
- gmisclib.g_lin_fit.test3 (*function*), 183
- gmisclib.g_lin_fit.test4 (*function*), 183
- gmisclib.g_lin_fit.test5 (*function*), 183
- gmisclib.g_lin_fit.test6 (*function*), 183
- gmisclib.g_localfit (*module*), 185–189
 - gmisclib.g_localfit.err_before_fit (*function*), 185
 - gmisclib.g_localfit.fit_giving_sigmas (*function*), 187
 - gmisclib.g_localfit.leaktest (*function*), 188
 - gmisclib.g_localfit.localfit (*function*), 185
 - gmisclib.g_localfit.pack (*function*), 185
 - gmisclib.g_localfit.reg_localfit (*function*), 186
 - gmisclib.g_localfit.robust_localfit (*function*), 187
 - gmisclib.g_localfit.test_fgs1 (*function*), 188
 - gmisclib.g_localfit.test_localfit11 (*function*), 188
 - gmisclib.g_localfit.test_localfit11e (*function*), 188
 - gmisclib.g_localfit.test_localfit21 (*function*), 188
 - gmisclib.g_localfit.test_localfit21u (*function*), 188
 - gmisclib.g_localfit.test_localfit22 (*function*), 188
 - gmisclib.g_localfit.test_wt (*function*), 188
- gmisclib.g_lpc (*module*), 190
 - gmisclib.g_lpc.add_sloppy (*function*), 190
 - gmisclib.g_lpc.lp1_fit (*function*), 190
 - gmisclib.g_lpc.lp1_run (*function*), 190
- gmisclib.g_pipe (*module*), 191–192
 - gmisclib.g_pipe.pfd (*class*), 191–192
 - gmisclib.g_pipe.popen2 (*function*), 191
 - gmisclib.g_pipe.test (*function*), 191
- gmisclib.g_pipe_old (*module*), 193–194
 - gmisclib.g_pipe_old.pfd (*class*), 193–194
 - gmisclib.g_pipe_old.popen2 (*function*), 193
 - gmisclib.g_pipe_old.test (*function*), 193
- gmisclib.g_place_label (*module*), 195–198
 - gmisclib.g_place_label.boxc (*class*), 195–196
 - gmisclib.g_place_label.Gauss (*function*), 195
 - gmisclib.g_place_label.gauss (*function*), 195
 - gmisclib.g_place_label.q-gauss (*function*), 195
 - gmisclib.g_place_label.text_template (*class*), 196–197
 - gmisclib.g_place_label.viewport (*class*), 197–198
- gmisclib.g_pylab (*module*), 199
- gmisclib.g_selector (*module*), 200–204
 - gmisclib.g_selector.false (*class*), 201–202
 - gmisclib.g_selector.one_selector (*class*), 203–204
 - gmisclib.g_selector.selector (*class*), 200–201
 - gmisclib.g_selector.selector_not (*function*), 200
 - gmisclib.g_selector.selector_op (*class*), 202–203
 - gmisclib.g_selector.test (*function*), 200
 - gmisclib.g_selector.true (*class*), 201
- gmisclib.gpk_getopt (*module*), 205
 - gmisclib.gpk_getopt.parse (*function*), 205
- gmisclib.gpk_hdr (*module*), 206
 - gmisclib.gpk_hdr.hdr (*class*), 206
 - gmisclib.gpk_hdr.uq (*function*), 206
- gmisclib.gpk_lapack (*module*), 207–208
 - gmisclib.gpk_lapack.err (*function*), 207
 - gmisclib.gpk_lapack.NoSolutionError (*class*), 207–208
 - gmisclib.gpk_lapack.solve (*function*), 207
 - gmisclib.gpk_lapack.test1 (*function*), 207
 - gmisclib.gpk_lapack.test2 (*function*), 207
 - gmisclib.gpk_lapack.testa (*function*), 207
 - gmisclib.gpk_lapack.testbdi (*function*), 207
- gmisclib.gpk_ls (*module*), 209–215
 - gmisclib.gpk_ls.all_between (*function*), 209
 - gmisclib.gpk_ls.linear_least_squares (*class*), 210–213

- gmisclib.gpk_lsq.lls_base (*class*), 209–210
- gmisclib.gpk_lsq.reg_linear_least_squares (*class*), 213–215
- gmisclib.gpk_lsq.test_hat (*function*), 209
- gmisclib.gpk_lsq.test_m1 (*function*), 209
- gmisclib.gpk_lsq.test_m2 (*function*), 209
- gmisclib.gpk_lsq.test_m2r (*function*), 209
- gmisclib.gpk_lsq.test_m2rR (*function*), 209
- gmisclib.gpk_lsq.test_svd (*function*), 209
- gmisclib.gpk_lsq.test_vec (*function*), 209
- gmisclib.gpk_lsq.test_vec2 (*function*), 209
- gmisclib.gpk_rlsq (*module*), 216–222
 - gmisclib.gpk_rlsq.NotEnoughData (*class*), 216–217
 - gmisclib.gpk_rlsq.robust_linear_fit (*class*), 220–222
 - gmisclib.gpk_rlsq.test_r1 (*function*), 216
 - gmisclib.gpk_rlsq.test_r1hat (*function*), 216
 - gmisclib.gpk_rlsq.test_w1 (*function*), 216
 - gmisclib.gpk_rlsq.test_w1b (*function*), 216
 - gmisclib.gpk_rlsq.w_linear_least_squares (*class*), 217–220
- gmisclib.gpk_writer (*module*), 223–225
 - gmisclib.gpk_writer.null_writer (*class*), 224–225
 - gmisclib.gpk_writer.writer (*class*), 223–224
- gmisclib.gpkequations (*module*), 226–229
 - gmisclib.gpkequations.add_slope_curv_droop (*function*), 226
 - gmisclib.gpkequations.add_something (*function*), 226
 - gmisclib.gpkequations.BadParamError (*class*), 227–228
 - gmisclib.gpkequations.bandwidth (*function*), 226
 - gmisclib.gpkequations.build_matrices (*function*), 226
 - gmisclib.gpkequations.eqn_c (*class*), 229
 - gmisclib.gpkequations.full_matrices (*function*), 226
 - gmisclib.gpkequations.NoTermsEx (*class*), 228–229
 - gmisclib.gpkequations.range_of_indices (*function*), 226
 - gmisclib.gpkequations.solve_eqns (*function*), 226
 - gmisclib.gpkequations.test (*function*), 226
 - gmisclib.gpkequations.test_as (*function*), 226
 - gmisclib.gpkequations.test_bw (*function*), 226
 - gmisclib.gpkequations.test_eqn (*function*), 226
 - gmisclib.gpkequations.test_solve (*function*), 226
 - gmisclib.gpkequations.test_solve2 (*function*), 226
- gmisclib.gpkmisc (*module*), 230–238
 - gmisclib.gpkmisc.a_factor (*function*), 232
 - gmisclib.gpkmisc.asinh (*function*), 234
 - gmisclib.gpkmisc.avg (*function*), 230
 - gmisclib.gpkmisc.chooseP (*function*), 235
 - gmisclib.gpkmisc.ComplexMedian (*function*), 231
 - gmisclib.gpkmisc.dir_lock (*class*), 236–237
 - gmisclib.gpkmisc.dropfront (*function*), 232
 - gmisclib.gpkmisc.entropy (*function*), 230
 - gmisclib.gpkmisc.erf (*function*), 234
 - gmisclib.gpkmisc.factor (*function*), 232
 - gmisclib.gpkmisc.find_in_PATH (*function*), 233
 - gmisclib.gpkmisc.gammaln (*function*), 232
 - gmisclib.gpkmisc.gcd (*function*), 233
 - gmisclib.gpkmisc.geo_mean (*function*), 230
 - gmisclib.gpkmisc.get_mtime (*function*), 233
 - gmisclib.gpkmisc.jackknife (*function*), 230
 - gmisclib.gpkmisc.log_Combinations (*function*), 231
 - gmisclib.gpkmisc.log_factorial (*function*), 231
 - gmisclib.gpkmisc.log_Student_t_dens (*function*), 230
 - gmisclib.gpkmisc.makedirs (*function*), 231
 - gmisclib.gpkmisc.mean_ad (*function*), 230

- gmisclib.gpkmisc.median (*function*), 230
- gmisclib.gpkmisc.median_ad (*function*), 230
- gmisclib.gpkmisc.need_to_recompute (*function*), 234
- gmisclib.gpkmisc.open_compressed (*function*), 232
- gmisclib.gpkmisc.open_nowipe (*function*), 231
- gmisclib.gpkmisc.prereq_mtime (*function*), 234
- gmisclib.gpkmisc.PrereqError (*class*), 237–238
- gmisclib.gpkmisc.primes (*function*), 232
- gmisclib.gpkmisc.resample (*function*), 230
- gmisclib.gpkmisc.shuffle_Nrep (*function*), 231
- gmisclib.gpkmisc.Student_t_dens (*function*), 230
- gmisclib.gpkmisc.test_primes (*function*), 233
- gmisclib.gpkmisc.testCM (*function*), 231
- gmisclib.gpkmisc.testSNR (*function*), 231
- gmisclib.gpkmisc.thr_iter_read (*function*), 231
- gmisclib.gpkmisc.threaded_readable_file (*class*), 235–236
- gmisclib.gpkmisc.truncate (*function*), 234
- gmisclib.hilbert_xform (*module*), 239
 - gmisclib.hilbert_xform.hilbert (*function*), 239
 - gmisclib.hilbert_xform.nextpow2 (*function*), 239
- gmisclib.HList (*module*), 29
 - gmisclib.HList.read (*function*), 29
- gmisclib.HTK_HMM_io (*module*), 30–39
 - gmisclib.HTK_HMM_io.BadFormatError (*class*), 30–31
 - gmisclib.HTK_HMM_io.deref (*function*), 30
 - gmisclib.HTK_HMM_io.dict_format (*function*), 30
 - gmisclib.HTK_HMM_io.hmm (*class*), 35–36
 - gmisclib.HTK_HMM_io.hmmset (*class*), 36–37
 - gmisclib.HTK_HMM_io.HTKformat (*function*), 30
 - gmisclib.HTK_HMM_io.mixture (*class*), 33–34
 - gmisclib.HTK_HMM_io.NoPhonemeToFormat (*class*), 38–39
 - gmisclib.HTK_HMM_io.read (*function*), 30
 - gmisclib.HTK_HMM_io.read_floats_b (*function*), 30
 - gmisclib.HTK_HMM_io.read_floats_t (*function*), 30
 - gmisclib.HTK_HMM_io.read_vec (*function*), 30
 - gmisclib.HTK_HMM_io.ref (*class*), 34
 - gmisclib.HTK_HMM_io.state (*class*), 34–35
 - gmisclib.HTK_HMM_io.test (*function*), 30
 - gmisclib.HTK_HMM_io.tokstream (*class*), 37–38
 - gmisclib.HTK_HMM_io.vec (*class*), 31–33
- gmisclib.HTK_MLF_io (*module*), 40–44
 - gmisclib.HTK_MLF_io.parse_label_line (*function*), 40
 - gmisclib.HTK_MLF_io.read (*function*), 41
 - gmisclib.HTK_MLF_io.readiter (*function*), 40
 - gmisclib.HTK_MLF_io.readone (*function*), 40
 - gmisclib.HTK_MLF_io.ReferencedFileNotFound (*class*), 42–43
 - gmisclib.HTK_MLF_io.writer (*class*), 43–44
- gmisclib.kl_dist (*module*), 240–243
 - gmisclib.kl_dist.cross (*function*), 241
 - gmisclib.kl_dist.estimate_tr_probs (*function*), 240
 - gmisclib.kl_dist.kl_nonzero_prob_m (*function*), 240

- gmisclib.kl_dist.kl_nonzero_probs (*function*), 240
- gmisclib.kl_dist.kl_nonzero_tr_prob_m (*function*), 241
- gmisclib.kl_dist.kl_nonzero_tr_probs (*function*), 241
- gmisclib.kl_dist.kldist_Markov (*function*), 241
- gmisclib.kl_dist.kldist_Markov_m (*function*), 241
- gmisclib.kl_dist.kldist_Markov_mm (*function*), 241
- gmisclib.kl_dist.kldist_vec (*function*), 240
- gmisclib.kl_dist.multinomial_fixer (*function*), 240
- gmisclib.kl_dist.multinomial_logp (*function*), 240
- gmisclib.kl_dist.NoConvergenceError (*class*), 241–242
- gmisclib.kl_dist.NotMarkovError (*class*), 242–243
- gmisclib.kl_dist.P (*function*), 240
- gmisclib.kl_dist.solve_for_pi (*function*), 241
- gmisclib.kl_dist.tr_from_obs (*function*), 240
- gmisclib.lapack_dgesv (*module*), 244
 - gmisclib.lapack_dgesv.dgesv (*function*), 244
- gmisclib.lapack_dsyevd (*module*), 245
 - gmisclib.lapack_dsyevd.dsyevd (*function*), 245
- gmisclib.load_mod (*module*), 246–248
 - gmisclib.load_mod.load (*function*), 246
 - gmisclib.load_mod.load_named (*function*), 247, 248
 - gmisclib.load_mod.split_name (*function*), 246
- gmisclib.lreg_fill (*module*), 249
 - gmisclib.lreg_fill.fill (*function*), 249
- gmisclib.makemake (*module*), 250–252
 - gmisclib.makemake.blank (*function*), 250
 - gmisclib.makemake.date (*function*), 250
 - gmisclib.makemake.FileNotFound (*class*), 251–252
 - gmisclib.makemake.finish (*function*), 250
 - gmisclib.makemake.log (*function*), 250
 - gmisclib.makemake.maketemp (*function*), 250
 - gmisclib.makemake.ncpu (*function*), 250
 - gmisclib.makemake.path_to (*function*), 250
 - gmisclib.makemake.quote (*function*), 250
 - gmisclib.makemake.read (*function*), 251
 - gmisclib.makemake.rule (*function*), 250
 - gmisclib.makemake.set_debug (*function*), 250
 - gmisclib.makemake.set_make_prog (*function*), 250
 - gmisclib.makemake.setlog (*function*), 250
 - gmisclib.makemake.var (*function*), 250
- gmisclib.matrix_arrange (*module*), 253
 - gmisclib.matrix_arrange.diagonalize (*function*), 253
 - gmisclib.matrix_arrange.map_array (*function*), 253
 - gmisclib.matrix_arrange.test (*function*), 253
- gmisclib.matrix_arrange_entropy (*module*), 254
 - gmisclib.matrix_arrange_entropy.make_min_entropy (*function*), 254
 - gmisclib.matrix_arrange_entropy.test2 (*function*), 254
 - gmisclib.matrix_arrange_entropy.test3 (*function*), 254
- gmisclib.matrix_rearrange_labels (*module*), 255–262
 - gmisclib.matrix_rearrange_labels.blockfom (*class*), 262
 - gmisclib.matrix_rearrange_labels.diagfom (*class*), 260–261
 - gmisclib.matrix_rearrange_labels.diagfom2 (*class*), 261–262
 - gmisclib.matrix_rearrange_labels.neg_near_diag2 (*function*), 258
 - gmisclib.matrix_rearrange_labels.pos_near_diag2 (*function*), 258
 - gmisclib.matrix_rearrange_labels.swap_toward_blocks

- (*function*), 259
- gmisclib.matrix_rearrange_labels.swap_toward_block (*function*), 258
- gmisclib.matrix_rearrange_labels.swap_toward_diag (*function*), 255
- gmisclib.matrix_rearrange_labels.swap_toward_diag3 (*function*), 257
- gmisclib.matrix_rearrange_labels.swap_toward_minimal_fom (*function*), 255
- gmisclib.matrix_rearrange_labels.swap_toward_positive (*function*), 256
- gmisclib.matrix_rearrange_labels.symm_swap_toward_block (*function*), 255
- gmisclib.matrix_rearrange_labels.test (*function*), 260
- gmisclib.matrix_rearrange_labels.test1 (*function*), 259
- gmisclib.mcmc (*module*), 263–289
 - gmisclib.mcmc.adjuster (*class*), 279–280
 - gmisclib.mcmc.Archive (*class*), 283–285
 - gmisclib.mcmc.BootStepper (*class*), 287–289
 - gmisclib.mcmc.bootstepper (*function*), 263
 - gmisclib.mcmc.ContPrmArchive (*class*), 285–287
 - gmisclib.mcmc.diag_variance (*function*), 264
 - gmisclib.mcmc.hashcounter_c (*class*), 282–283
 - gmisclib.mcmc.make_list_of_positions (*function*), 263
 - gmisclib.mcmc.NoBoot (*class*), 280–281
 - gmisclib.mcmc.NotGoodPosition (*class*), 281–282
 - gmisclib.mcmc.position_base (*class*), 267–269
 - gmisclib.mcmc.position_history_dependent (*class*), 273–276
 - gmisclib.mcmc.position_nonrepeatable (*class*), 271–273
 - gmisclib.mcmc.position_repeatable (*class*), 269–271
 - gmisclib.mcmc.problem_definition (*class*), 264–267
- gmisclib.mcmc.start_is_list_a (*function*), 263
- gmisclib.mcmc.start_is_list_p (*function*), 263
- gmisclib.mcmc.stepper (*class*), 277–279
- gmisclib.mcmc.T_acceptor (*class*), 276–277
- gmisclib.mcmc.test (*function*), 264
- gmisclib.mcmc_big (*module*), 304–307
 - gmisclib.mcmc_big.BootStepper (*class*), 304–307
 - gmisclib.mcmc_big.bootstepper (*function*), 304
 - gmisclib.mcmc_big.find_closest_p (*function*), 304
 - gmisclib.mcmc_big.pairmax (*function*), 304
 - gmisclib.mcmc_big.test (*function*), 304
- gmisclib.mcmc_cooperate (*module*), 308–313
 - gmisclib.mcmc_cooperate.Barrier (*class*), 308–309
 - gmisclib.mcmc_cooperate.connection (*class*), 311–313
 - gmisclib.mcmc_cooperate.LateToBarrier (*class*), 310–311
 - gmisclib.mcmc_cooperate.Oops (*class*), 309–310
 - gmisclib.mcmc_cooperate.op_float_median (*function*), 308
 - gmisclib.mcmc_cooperate.op_string_median (*function*), 308
 - gmisclib.mcmc_cooperate.test0 (*function*), 308
 - gmisclib.mcmc_cooperate.test0s (*function*), 308
 - gmisclib.mcmc_cooperate.test1 (*function*), 308
 - gmisclib.mcmc_cooperate.test_many (*function*), 308
- gmisclib.mcmc_helper (*module*), 314–322
 - gmisclib.mcmc_helper.logger_template (*class*), 316

- gmisclib.mcmc_helper.make_stepper_from_lov (function), 314
- gmisclib.mcmc_helper.step_acceptor (class), 320–322
- gmisclib.mcmc_helper.stepper (class), 316–320
- gmisclib.mcmc_helper.test (function), 314
- gmisclib.mcmc_helper.TooManyLoops (class), 314–315
- gmisclib.mcmc_helper.warnevery (class), 315–316
- gmisclib.mcmc_idxr (module), 323–331
 - gmisclib.mcmc_idxr.Expo (class), 325–327
 - gmisclib.mcmc_idxr.LogNormal (class), 329–331
 - gmisclib.mcmc_idxr.logp_prior_normalized (function), 323
 - gmisclib.mcmc_idxr.Normal (class), 327–328
 - gmisclib.mcmc_idxr.probdist_c (class), 323–324
 - gmisclib.mcmc_idxr.Uniform (class), 328–329
 - gmisclib.mcmc_idxr.Weibull (class), 324–325
- gmisclib.mcmc_logger (module), 332–339
 - gmisclib.mcmc_logger.logger_A (class), 334–336
 - gmisclib.mcmc_logger.logger_c (class), 333–334
 - gmisclib.mcmc_logger.logger_N (class), 336–337
 - gmisclib.mcmc_logger.lti_c (class), 337–338
 - gmisclib.mcmc_logger.NoDataError (class), 338–339
 - gmisclib.mcmc_logger.read_multisample (function), 332
 - gmisclib.mcmc_logger.WildNumber (class), 332–333
- gmisclib.mcmc_m4p (module), 340–344
 - gmisclib.mcmc_m4p.is_root (function), 340
 - gmisclib.mcmc_m4p.precompute_logp (function), 340
 - gmisclib.mcmc_m4p.rank (function), 340
 - gmisclib.mcmc_m4p.size (function), 340
 - gmisclib.mcmc_m4p.stepper (class), 340–344
 - gmisclib.mcmc_m4p.test (function), 340
- gmisclib.mcmc_mpi (module), 345–349
 - gmisclib.mcmc_mpi.precompute_logp (function), 345
 - gmisclib.mcmc_mpi.stepper (class), 345–349
 - gmisclib.mcmc_mpi.test (function), 345
- gmisclib.mcmc_pypar (module), 350–353
 - gmisclib.mcmc_pypar.is_root (function), 350
 - gmisclib.mcmc_pypar.precompute_logp (function), 350
 - gmisclib.mcmc_pypar.Reduce (function), 350
 - gmisclib.mcmc_pypar.size (function), 350
 - gmisclib.mcmc_pypar.stepper (class), 350–353
 - gmisclib.mcmc_pypar.test (function), 350
- gmisclib.mcmc_socket (module), 354–358
 - gmisclib.mcmc_socket.precompute_logp (function), 354
 - gmisclib.mcmc_socket.stepper (class), 354–358
 - gmisclib.mcmc_socket.test (function), 354
- gmisclib.mcmcS2 (module), 290
 - gmisclib.mcmcS2.go_step (function), 290
- gmisclib.mcmcSQ (module), 291–303
 - gmisclib.mcmcSQ.archiver (class), 302–303
 - gmisclib.mcmcSQ.BadlyConditioned (class), 295–296
 - gmisclib.mcmcSQ.CantGetPoint (class), 292–293
 - gmisclib.mcmcSQ.diag_variance (function), 291
 - gmisclib.mcmcSQ.DontGoHere (class), 293–294
 - gmisclib.mcmcSQ.get_1_localfit (function), 291

- gmisclib.mcmcSQ.get_localfit (*function*), 291
- gmisclib.mcmcSQ.go (*function*), 291
- gmisclib.mcmcSQ.Lambda_c (*class*), 298–300
- gmisclib.mcmcSQ.linear_least_squares_lambda (*function*), 291
- gmisclib.mcmcSQ.NegativeEigenvalue (*class*), 294–295
- gmisclib.mcmcSQ.position (*class*), 297–298
- gmisclib.mcmcSQ.problem_definition (*class*), 296–297
- gmisclib.mcmcSQ.sigma (*function*), 291
- gmisclib.mcmcSQ.SQstepper (*class*), 300–302
- gmisclib.mcmcSQ.test (*function*), 291
- gmisclib.mcmcSQ.test_get_localfit (*function*), 291
- gmisclib.mcmcSQ.test_llsl (*function*), 291
- gmisclib.mcmcSQ.test_llsl_1 (*function*), 291
- gmisclib.mcmcSQ.test_llsl_1o (*function*), 291
- gmisclib.mcmcSQ.test_llsl_2 (*function*), 291
- gmisclib.mcmcSQ.test_llsl_2u (*function*), 291
- gmisclib.mcmcSQ.test_stepper (*function*), 291
- gmisclib.mcmcSQ.timer (*class*), 296
- gmisclib.md_scaling (*module*), 359
 - gmisclib.md_scaling.distances (*function*), 359
 - gmisclib.md_scaling.find_rep (*function*), 359
 - gmisclib.md_scaling.test (*function*), 359
- gmisclib.MFCCFile (*module*), 45
 - gmisclib.MFCCFile.MFCCFile (*class*), 45
- gmisclib.MLF_file (*module*), 46–49
 - gmisclib.MLF_file.BadFormatError (*class*), 46–47
 - gmisclib.MLF_file.block_MLF_file (*class*), 48–49
 - gmisclib.MLF_file.NotInMLFFile (*class*), 47–48
- gmisclib.multivariate_classes (*module*), 360–363
 - gmisclib.multivariate_classes.diag_inv_variance (*function*), 360
 - gmisclib.multivariate_classes.model_with_numbers (*class*), 362–363
 - gmisclib.multivariate_classes.modeldesc (*class*), 361–362
 - gmisclib.multivariate_classes.QuadraticNotNormaliz (*class*), 360–361
 - gmisclib.multivariate_classes.vec_inv_variance (*function*), 360
- gmisclib.multivariate_mm (*module*), 364–369
 - gmisclib.multivariate_mm._multi_mu_init_ (*function*), 364
 - gmisclib.multivariate_mm.datum_c (*class*), 364
 - gmisclib.multivariate_mm.multi_mu (*class*), 364–366
 - gmisclib.multivariate_mm.multi_mu_diag (*class*), 367–368
 - gmisclib.multivariate_mm.multi_mu_diag_with_number (*class*), 368–369
 - gmisclib.multivariate_mm.multi_mu_with_numbers (*class*), 366–367
- gmisclib.multivariate_q (*module*), 370–375
 - gmisclib.multivariate_q.diag_quadratic (*class*), 372–373
 - gmisclib.multivariate_q.diag_quadratic_with_number (*class*), 373–375
 - gmisclib.multivariate_q.quadratic (*class*), 370–371
 - gmisclib.multivariate_q.quadratic_with_numbers (*class*), 371–372
- gmisclib.multivariate_normal (*module*), 376
 - gmisclib.multivariate_normal.multivariate_normal (*class*), 376
 - gmisclib.multivariate_normal.test (*function*), 376

- gmisclib.named_block_file (*module*), 377
 - gmisclib.named_block_file.read (*function*), 377
 - gmisclib.named_block_file.test (*function*), 377
 - gmisclib.named_block_file.write (*function*), 377
 - gmisclib.named_block_file.write_key (*function*), 377
 - gmisclib.named_block_file.write_line (*function*), 377
 - gmisclib.named_block_file.write_line_kv (*function*), 377
 - gmisclib.named_block_file.write_text (*function*), 377
- gmisclib.nbest (*module*), 378–379
 - gmisclib.nbest.go (*function*), 378
 - gmisclib.nbest.node (*class*), 378–379
 - gmisclib.nbest.test (*function*), 378
- gmisclib.nice_hash (*module*), 380–388
 - gmisclib.nice_hash.DontHashThis (*class*), 380–381
 - gmisclib.nice_hash.hash (*class*), 383–388
 - gmisclib.nice_hash.map (*function*), 380
 - gmisclib.nice_hash.NotInHash (*class*), 381–382
 - gmisclib.nice_hash.simple (*class*), 382–383
- gmisclib.nicknames (*module*), 389
 - gmisclib.nicknames.nicknames (*function*), 389
 - gmisclib.nicknames.test (*function*), 389
- gmisclib.nmf (*module*), 390
 - gmisclib.nmf.nmf (*function*), 390
- gmisclib.Num (*module*), 50–54
- gmisclib.Numeric_gpk (*module*), 55–60
 - gmisclib.Numeric_gpk.add_overlap (*function*), 55
 - gmisclib.Numeric_gpk.argmax (*function*), 55
 - gmisclib.Numeric_gpk.bevel_concat (*function*), 55
 - gmisclib.Numeric_gpk.block_stdev (*function*), 57
 - gmisclib.Numeric_gpk.convolve (*function*), 57
 - gmisclib.Numeric_gpk.edge_window (*function*), 57
 - gmisclib.Numeric_gpk.edge_window_t (*function*), 58
 - gmisclib.Numeric_gpk.EdgesTooWide (*class*), 59–60
 - gmisclib.Numeric_gpk.interp (*function*), 56
 - gmisclib.Numeric_gpk.interpN (*function*), 56
 - gmisclib.Numeric_gpk.KolmogorovSmirnov (*function*), 56
 - gmisclib.Numeric_gpk.limit (*function*), 56
 - gmisclib.Numeric_gpk.median_across (*function*), 55
 - gmisclib.Numeric_gpk.N_frac_rank (*function*), 55
 - gmisclib.Numeric_gpk.N_maximum (*function*), 55
 - gmisclib.Numeric_gpk.N_mean_ad (*function*), 55
 - gmisclib.Numeric_gpk.N_median (*function*), 55
 - gmisclib.Numeric_gpk.N_median_across (*function*), 56
 - gmisclib.Numeric_gpk.N_minimum (*function*), 55
 - gmisclib.Numeric_gpk.Poisson (*function*), 55
 - gmisclib.Numeric_gpk.qform (*function*), 56
 - gmisclib.Numeric_gpk.set_diag (*function*), 56
 - gmisclib.Numeric_gpk.split_into_clumps (*function*), 56
 - gmisclib.Numeric_gpk.trimmed_mean_across (*function*), 56
 - gmisclib.Numeric_gpk.trimmed_mean_sigma_across (*function*), 56
 - gmisclib.Numeric_gpk.trimmed_stdev_across (*function*), 56

- gmisclib.Numeric_gpk.vec_variance (*function*), 56
- gmisclib.Numeric_gpk.zero_pad_end (*function*), 55
- gmisclib.opt (*module*), 391–404
 - gmisclib.opt.anneal_guts (*function*), 392
 - gmisclib.opt.BadParamError (*class*), 397–398
 - gmisclib.opt.BadResult (*class*), 398–399
 - gmisclib.opt.clz_point (*function*), 391
 - gmisclib.opt.deriv_estimate (*function*), 392
 - gmisclib.opt.diff_one (*function*), 392
 - gmisclib.opt.eval_lambda (*function*), 392
 - gmisclib.opt.explored_region (*function*), 391
 - gmisclib.opt.lamb_correct (*function*), 392
 - gmisclib.opt.linconst_max (*function*), 393
 - gmisclib.opt.linconst_min (*function*), 393
 - gmisclib.opt.linear_constraint (*function*), 393
 - gmisclib.opt.LockedList (*class*), 401
 - gmisclib.opt.maxabs (*function*), 391
 - gmisclib.opt.mysem (*class*), 399–400
 - gmisclib.opt.near_duplicate (*function*), 391
 - gmisclib.opt.need_more_diff_pts (*function*), 392
 - gmisclib.opt.NoDerivative (*class*), 396–397
 - gmisclib.opt.NoDownhill (*class*), 395–396
 - gmisclib.opt.opt (*class*), 401–404
 - gmisclib.opt.OptError (*class*), 394–395
 - gmisclib.opt.prms (*class*), 400–401
 - gmisclib.opt.scale_est (*function*), 391
 - gmisclib.opt.semclass (*class*), 400
 - gmisclib.opt.sumsq (*function*), 391
 - gmisclib.opt.symmetry_point (*function*), 391
 - gmisclib.opt.test1 (*function*), 392
 - gmisclib.opt.test1_fcn (*function*), 392
 - gmisclib.opt.test1a (*function*), 392
 - gmisclib.opt.test1a_fcn (*function*), 392
 - gmisclib.opt.test2 (*function*), 392
 - gmisclib.opt.test2_fcn (*function*), 392
 - gmisclib.opt.test3 (*function*), 392
 - gmisclib.opt.test3_fcn (*function*), 392
 - gmisclib.opt.test4 (*function*), 393
 - gmisclib.opt.test4_fcn (*function*), 393
 - gmisclib.opt.test5 (*function*), 393
 - gmisclib.opt.test5_fcn (*function*), 393
 - gmisclib.opt.test6 (*function*), 393
 - gmisclib.opt.test6_fcn (*function*), 393
 - gmisclib.opt.test7 (*function*), 393
 - gmisclib.opt.vec_equal (*function*), 391
 - gmisclib.opt.wtf (*function*), 391
- gmisclib.ortho_poly (*module*), 405–415
 - gmisclib.ortho_poly.Chebyshev (*class*), 409–411
 - gmisclib.ortho_poly.Chebyshev2 (*class*), 411–412
 - gmisclib.ortho_poly.F (*function*), 405
 - gmisclib.ortho_poly.Legendre (*class*), 408–409
 - gmisclib.ortho_poly.ortho (*class*), 405–406
 - gmisclib.ortho_poly.ortho_poly (*class*), 406–408
 - gmisclib.ortho_poly.SinCos (*class*), 412–413
 - gmisclib.ortho_poly.SLTB (*class*), 413–415
 - gmisclib.ortho_poly.test (*function*), 405
- gmisclib.parse_tree_number (*module*), 416–435
 - gmisclib.parse_tree_number.abs (*function*), 416
 - gmisclib.parse_tree_number.abstract_number (*class*), 419–420
 - gmisclib.parse_tree_number.Array (*class*), 416–417
 - gmisclib.parse_tree_number.coerce_into (*function*), 416
 - gmisclib.parse_tree_number.cos (*function*), 416
 - gmisclib.parse_tree_number.elementof (*class*), 430–432
 - gmisclib.parse_tree_number.exp (*function*), 416

- gmisclib.parse_tree_number.Expression (*class*), 420–423
- gmisclib.parse_tree_number.Float (*class*), 425–427
- gmisclib.parse_tree_number.is_mul_of (*function*), 416
- gmisclib.parse_tree_number.log (*function*), 416
- gmisclib.parse_tree_number.Name (*class*), 423–425
- gmisclib.parse_tree_number.operator1 (*class*), 427–430
- gmisclib.parse_tree_number.operatorN (*function*), 416
- gmisclib.parse_tree_number.operatorNg1 (*class*), 432–435
- gmisclib.parse_tree_number.output_mixin (*class*), 417–419
- gmisclib.parse_tree_number.sin (*function*), 416
- gmisclib.parse_tree_number.sqrt (*function*), 416
- gmisclib.permute (*module*), 436
 - gmisclib.permute.factorial (*function*), 436
 - gmisclib.permute.next (*function*), 436
 - gmisclib.permute.permutations (*function*), 436
- gmisclib.pylab_oneaxis (*module*), 437
 - gmisclib.pylab_oneaxis.plot_oneaxis (*function*), 437
- gmisclib.pylab_starplot (*module*), 438–440
 - gmisclib.pylab_starplot.errorbar (*class*), 439–440
 - gmisclib.pylab_starplot.errorbar_maker (*class*), 438–439
 - gmisclib.pylab_starplot.grey_out (*function*), 438
 - gmisclib.pylab_starplot.plot (*function*), 438
 - gmisclib.pylab_starplot.star (*function*), 438
- gmisclib.read_dicom (*module*), 441–442
 - gmisclib.read_dicom.img_with_mx (*class*), 441–442
- gmisclib.read_dicom.read_body (*function*), 441
- gmisclib.read_dicom.read_header (*function*), 441
- gmisclib.read_dicom.read_imgs (*function*), 441
- gmisclib.resample (*module*), 443
 - gmisclib.resample.compare (*function*), 443
 - gmisclib.resample.simple (*function*), 443
 - gmisclib.resample.test (*function*), 443
- gmisclib.robust_multivariate (*module*), 444
 - gmisclib.robust_multivariate.cov_wt (*function*), 444
 - gmisclib.robust_multivariate.covariance (*function*), 444
 - gmisclib.robust_multivariate.integrate (*function*), 444
 - gmisclib.robust_multivariate.integrate_guts (*function*), 444
 - gmisclib.robust_multivariate.wtdim (*function*), 444
 - gmisclib.robust_multivariate.wtfunc (*function*), 444
- gmisclib.root (*module*), 445
 - gmisclib.root.iroot (*function*), 445
 - gmisclib.root.root (*function*), 445
 - gmisclib.root.test (*function*), 445
 - gmisclib.root.testi (*function*), 445
- gmisclib.rubber_array (*module*), 446–447
 - gmisclib.rubber_array.ext_block (*class*), 446
 - gmisclib.rubber_array.extensible_array (*class*), 446–447
- gmisclib.s_lin_fit (*module*), 448–450
 - gmisclib.s_lin_fit.Error (*class*), 448–449
 - gmisclib.s_lin_fit.NoDataError (*class*), 449–450
 - gmisclib.s_lin_fit.plane (*function*), 448
 - gmisclib.s_lin_fit.test (*function*), 448
- gmisclib.sbd_array (*module*), 451–453
 - gmisclib.sbd_array.err (*function*), 451
 - gmisclib.sbd_array.multiply (*function*), 451
 - gmisclib.sbd_array.NoSolutionError (*class*),

- 452–453
- gmisclib.sbd_array.sbd (*class*), 451–452
- gmisclib.sbd_array.solve (*function*), 451
- gmisclib.sbd_array.symmetrize (*function*), 451
- gmisclib.sbd_array.test1 (*function*), 451
- gmisclib.sbd_array.test2 (*function*), 451
- gmisclib.sbd_array.testa (*function*), 451
- gmisclib.sbd_array.testbdi (*function*), 451
- gmisclib.sbd_array.testm (*function*), 451
- gmisclib.segmentfile (*module*), 454–455
 - gmisclib.segmentfile.group (*class*), 455
 - gmisclib.segmentfile.parse (*function*), 454
 - gmisclib.segmentfile.read (*function*), 454
 - gmisclib.segmentfile.segment (*class*), 454–455
 - gmisclib.segmentfile.test (*function*), 454
- gmisclib.sharp_energy (*module*), 456
 - gmisclib.sharp_energy.complex_median (*function*), 456
 - gmisclib.sharp_energy.one_median (*function*), 456
 - gmisclib.sharp_energy.pwr (*function*), 456
 - gmisclib.sharp_energy.stepped_median (*function*), 456
- gmisclib.solve_sum_abs (*module*), 457–458
 - gmisclib.solve_sum_abs.solve1 (*function*), 457
 - gmisclib.solve_sum_abs.solve_fit (*function*), 457
 - gmisclib.solve_sum_abs.solve_fit_wt (*function*), 457
 - gmisclib.solve_sum_abs.test1 (*function*), 458
 - gmisclib.solve_sum_abs.test2 (*function*), 458
- gmisclib.spread_jobs (*module*), 459–464
 - gmisclib.spread_jobs.append (*function*), 460
 - gmisclib.spread_jobs.main (*function*), 460
 - gmisclib.spread_jobs.main2 (*function*), 460
 - gmisclib.spread_jobs.NoResponse (*class*), 461–462
 - gmisclib.spread_jobs.notComputed (*class*), 461
 - gmisclib.spread_jobs.one_shot_test (*function*), 461
 - gmisclib.spread_jobs.RemoteException (*class*), 462–464
 - gmisclib.spread_jobs.Replace (*function*), 460
 - gmisclib.spread_jobs.replace (*function*), 460
 - gmisclib.spread_jobs.test (*function*), 461
 - gmisclib.spread_jobs.test_worker (*function*), 461
 - gmisclib.spread_jobs.unpickled_pseudofile (*class*), 464
- gmisclib.stats (*module*), 465–466
 - gmisclib.stats.betacf (*function*), 465
 - gmisclib.stats.betainc (*function*), 465
 - gmisclib.stats.f_value (*function*), 465
 - gmisclib.stats.fprob (*function*), 465
 - gmisclib.stats.gammaln (*function*), 466
 - gmisclib.stats.t_value (*function*), 466
 - gmisclib.stats.test_fprob (*function*), 466
- gmisclib.synch_demod (*module*), 467
 - gmisclib.synch_demod.expand (*function*), 467
 - gmisclib.synch_demod.one_ampl (*function*), 467
 - gmisclib.synch_demod.pwr (*function*), 467
 - gmisclib.synch_demod.pwr_guts (*function*), 467
 - gmisclib.synch_demod.pwrs (*function*), 467
 - gmisclib.synch_demod.stepped_ampl (*function*), 467
 - gmisclib.synch_demod.test0 (*function*), 467
 - gmisclib.synch_demod.test1 (*function*), 467
- gmisclib.threaded_io (*module*), 468–473
 - gmisclib.threaded_io.CaughtException (*class*), 468–469
 - gmisclib.threaded_io.check_n_raise (*function*), 468
 - gmisclib.threaded_io.ncpu (*function*), 468

- gmisclib.threaded_io.pairmap (*function*), 468
- gmisclib.threaded_io.testmap (*function*), 468
- gmisclib.threaded_io.Thread_poolR (*class*), 472–473
- gmisclib.threaded_io.Thread_poolW (*class*), 470–472
- gmisclib.threaded_io.threading_with_result (*class*), 469–470
- gmisclib.threaded_io.to_be_joined (*function*), 468
- gmisclib.tsops (*module*), 474–476
 - gmisclib.tsops.apply (*function*), 475
 - gmisclib.tsops.axis (*class*), 476
 - gmisclib.tsops.common (*function*), 475
 - gmisclib.tsops.commonN (*function*), 475
 - gmisclib.tsops.copy_interval (*function*), 475
 - gmisclib.tsops.interp (*function*), 474
 - gmisclib.tsops.interp_fill (*function*), 474
 - gmisclib.tsops.interpN (*function*), 474
 - gmisclib.tsops.interpN_fill (*function*), 474
 - gmisclib.tsops.mul (*function*), 475
 - gmisclib.tsops.resample (*function*), 475
 - gmisclib.tsops.test (*function*), 476
 - gmisclib.tsops.test_fig (*function*), 474
 - gmisclib.tsops.test_interp1 (*function*), 474
 - gmisclib.tsops.test_interp2 (*function*), 474
 - gmisclib.tsops.test_interp3 (*function*), 474
 - gmisclib.tsops.time (*function*), 474
 - gmisclib.tsops.time2 (*function*), 474
- gmisclib.wavesurfer_lab (*module*), 477–478
 - gmisclib.wavesurfer_lab.read (*function*), 477
 - gmisclib.wavesurfer_lab.write (*function*), 477
- gmisclib.wavio (*module*), 479–482
 - gmisclib.wavio.BadFileFormatError (*class*), 481–482
 - gmisclib.wavio.Overflow (*class*), 480–481
 - gmisclib.wavio.read (*function*), 479
 - gmisclib.wavio.read_hdr (*function*), 479
 - gmisclib.wavio.write (*function*), 479
- gmisclib.weighted_percentile (*module*), 483–484
 - gmisclib.weighted_percentile.test (*function*), 484
 - gmisclib.weighted_percentile.test_median (*function*), 484
 - gmisclib.weighted_percentile.wp (*function*), 483
 - gmisclib.weighted_percentile.wtd_median (*function*), 483
- gmisclib.xwaves_errs (*module*), 485–490
 - gmisclib.xwaves_errs.BadFileFormatError (*class*), 487–488
 - gmisclib.xwaves_errs.DataError (*class*), 488–489
 - gmisclib.xwaves_errs.DataOutOfOrderError (*class*), 489–490
 - gmisclib.xwaves_errs.Error (*class*), 485–486
 - gmisclib.xwaves_errs.NoSuchFileError (*class*), 486–487
- gmisclib.xwaves_lab (*module*), 491
 - gmisclib.xwaves_lab.end_marks (*function*), 491
 - gmisclib.xwaves_lab.read (*function*), 491
 - gmisclib.xwaves_lab.start_stop (*function*), 491
 - gmisclib.xwaves_lab.write (*function*), 491
- gmisclib.xwaves_mark (*module*), 492
 - gmisclib.xwaves_mark.combine_2labs (*function*), 492
 - gmisclib.xwaves_mark.mark_to_lab (*function*), 492
 - gmisclib.xwaves_mark.read (*function*), 492
 - gmisclib.xwaves_mark.write (*function*), 492
- gpk_wavio (*module*), 493
- Markov Chain Monte-Carlo, 317, 319, 342, 343, 347, 348, 351, 352, 356, 357
- mcmc (*module*), 494–496
 - mcmc.def_logger (*class*), 496
 - mcmc.def_logger.__init__ (*method*), 496
 - mcmc.def_logger.add (*method*), 496
 - mcmc.def_logger.finish (*method*), 496

- mcmc.go (*function*), 496
- mcmc.logp_from_resid (*function*), 496
- q3html (*module*), 497–500
 - q3html.aget (*function*), 498
 - q3html.agp (*function*), 499
 - q3html.av (*function*), 498
 - q3html.del_fin_sl (*function*), 499
 - q3html.dequote (*function*), 498
 - q3html.dpre (*function*), 499
 - q3html.easygo (*function*), 499
 - q3html.escape (*function*), 498
 - q3html.file_only (*function*), 499
 - q3html.format_dict (*function*), 498
 - q3html.get (*function*), 499
 - q3html.get_logical_line (*function*), 498
 - q3html.go (*function*), 499
 - q3html.lineC (*class*), 499–500
 - q3html.lineC.__init__ (*method*), 500
 - q3html.lineC.getarg (*method*), 500
 - q3html.measure_indent (*function*), 498
 - q3html.needcopy (*function*), 499
 - q3html.prepare_text (*function*), 498
 - q3html.prepend (*function*), 498
 - q3html.process (*function*), 498
 - q3html.process_tag (*function*), 498
 - q3html.qfiles (*function*), 498
 - q3html.readheader (*function*), 499
 - q3html.starts_with_a_tag (*function*), 498
 - q3html.sw (*function*), 498
 - q3html.swapend (*function*), 498
 - q3html.tokenize (*function*), 498
 - q3html.urlq (*function*), 498
- run_several (*module*), 501–503
 - run_several.get_loadavg (*function*), 502
 - run_several.get_ncpu (*function*), 502
 - run_several.mem_pressure (*function*), 502
 - run_several.parse_NP (*function*), 502
 - run_several.run_processes (*function*), 502
 - run_several.set_oom (*function*), 502
 - run_several.wait_until_unloaded (*function*), 502
 - run_several.write_msgs (*function*), 502
- select_fiat_entries (*module*), 504–506
 - select_fiat_entries.avio_read (*function*), 504
 - select_fiat_entries.fiat_read (*function*), 504
 - select_fiat_entries.process (*function*), 504
 - select_fiat_entries.writer (*class*), 504–506
 - select_fiat_entries.writer.globals (*method*), 505
- selector, 147
- xpaset, 157, 167