

Supplementary Materials: Computing the parameter values for the emergence of homochirality in complex networks

Andrés Montoya ^{1,‡} , Elkin Cruz ^{2,‡}  and Jesús Ágreda ^{2,‡,*} 

1 ¹ Departamento de Matemáticas, Universidad Nacional de Colombia; jamontoyaa@unal.edu.co

2 ² Departamento de Química, Universidad Nacional de Colombia; elacruzca@unal.edu.co ;
3 jagreda@unal.edu.co

4 * Correspondence: jagreda@unal.edu.co; Tel.: +57-1-3165000 Ext. 14431

5 ‡ These authors contributed equally to this work.

6 This document presents information that is either complementary to the main paper, or too long
7 to be in it. The document is divided into the following sections: where to find the code used in the
8 calculations, typical examples of the Listanalchem outputs for the models studied in the paper, and
9 the Iwamoto model under imperfect conditions figure, which is not in the main document.

10 1. The software developed and used in the paper

11 The algorithm to study the stability of the models proposed to explain the origin of biological
12 homochirality, and developed in the paper, was implemented in the computer software Listanalchem,
13 option six. The software is available at:

14 <http://ciencias.bogota.unal.edu.co/gruposdeinvestigacion/grupo-de-estudios-para-la-remediacion-y-mitigacion-de-impactos-negativos-al-ambiente-germina/software/>

16 or

17 <https://gitlab.com/homochirality>

18 The simulations of the models (the numerical solution of the respective ordinary differential
19 equations) were made using the software Chemkinlator, also available at the same URL given above.

20 2. Inputs and outputs of Listanalchem for the models studied in the paper

21 In the following subsections, a typical output example of Listanalchem, option six, is presented
22 for the four models studied in the paper. The use of random sampling, in some steps of the algorithm,
23 makes the numerical part of the outputs calculated by Listanalchem different every single time the
24 program is executed. Thus, if you run Listanalchem on the models studied in the paper, you may get
25 different quantitative results from those shown in figures 1, 2, and 3 of the paper. However, only the
26 quantitative aspects will be different, but not the qualitative ones.

27 Models in Listanalchem are defined as plain text files. A model must define a name, a set of
28 species, and a set of reactions. The models studied in the paper are in the folder “models” of the
29 program Listanalchem after it is installed. Please, see the README.md file, to get instructions to
30 install and run the program. The analysis of a network can be made writing the command line:

31 `python -m listanalchem --model models/Model-name.py`

32 In the next subsections, and for every model analyzed in the paper, the input and the output files
33 are presented.

34 2.1. The Calvin model

35 2.1.1. The input file

```
36 # run with: "python -m listanalchem --model models/Calvin-pseudoquiral.py"
37 modelname = 'Calvin - pseudoquiral'
38 species = ['L1', 'D1', 'L2', 'D2']
39 reactions = [
40     'L1 + L2 -> 2L2', # 0
41     'D1 + D2 -> 2D2', # 1
42     '2L2 -> L1 + L2', # 2
43     '2D2 -> D1 + D2', # 3
44     'L1 -> D1', # 4
45     'D1 -> L1', # 5
46     'L1 -> L2', # 6
47     'D1 -> D2', # 7
48     'L2 -> L1', # 8
49     'D2 -> D1', # 9
50 ]
51
52 dual_pairs = [(0, 1), (2, 3), (6, 7), (8, 9)]
53
54 analyses = {
55     "sna": {
56         "enabled": True,
57         "dual-pairs-in-ec": True,
58         # "instability-heuristic": "trace-determinant",
59         # "instability-heuristic": "characteristic-polynomial",
60         "instability-heuristic": "mineurs",
61         "sum-mineurs": True,
62         "max-mineur-search-stop": 5,
63         "simplification-tries": 1000,
64         "num-samples": 10,
65         "samples-folder": 'samples/calvin-pseudoquiral',
66     },
67     "frank-pseudoquiral": {
68         "enabled": True,
69         "enantiomeric-pairs": [(0, 1), (2, 3)],
70         "dual-pairs-in-ec": True,
71         # "instability-heuristic": "trace-determinant",
72         # "instability-heuristic": "characteristic-polynomial",
73         "instability-heuristic": "mineurs",
74         "sum-mineurs": True,
75         "max-mineur-search-stop": 5,
76         "simplification-tries": 1000,
77         "num-samples": 10,
78         "samples-folder": 'samples/calvin-pseudoquiral',
79     }
80 }
```

81 2.1.2. The output file

```
82 Welcome to Listanalchem!
83 Remember to check the model you're loading, it must:
84 * Have a name 'modelname'
85 * Have a list of species 'species' (note: the first two species will be considered enantiomers,
86   all other species do not)
87 * Have a non-empty list of reactions
88
89 *****
90 *** Calvin - pseudoquiral ***
91 *****
92
93 Species:
94 ['L1', 'D1', 'L2', 'D2']
95 Reactions list:
96 ['L1 + L2 -> 2 L2',
97  'D1 + D2 -> 2 D2',
98  '2 L2 -> L1 + L2',
99  '2 D2 -> D1 + D2',
100  'L1 -> D1',
101  'D1 -> L1',
102  'L1 -> L2',
103  'D1 -> D2',
104  'L2 -> L1',
105  'D2 -> D1']
106 Dual pair equations (equations are numbered from 0 to 9):
107 [(0, 1), (2, 3), (6, 7), (8, 9)]
108 Regular reactions (not dual)
109 [4, 5]
110
111
112 Stoichiometric Matrix:
113 [[-1  0  1  0 -1  1 -1  0  1  0]
114  [ 0 -1  0  1  1 -1  0 -1  0  1]
115  [ 1  0 -1  0  0  0  1  0 -1  0]
116  [ 0  1  0 -1  0  0  0  1  0 -1]]
117
118 Reactions Order Matrix:
119 [[1 0 0 0 1 0 1 0 0 0]
120  [0 1 0 0 0 1 0 1 0 0]
121  [1 0 2 0 0 0 0 0 1 0]
122  [0 1 0 2 0 0 0 0 0 1]]
123
124 Velocity Function:
125 [k0*x0*x2]
126 [          ]
127 [k1*x1*x3]
128 [          ]
129 [          2 ]
130 [ k2*x2  ]
```

```

131 [      ]
132 [      2 ]
133 [ k3*x3 ]
134 [      ]
135 [ k4*x0 ]
136 [      ]
137 [ k5*x1 ]
138 [      ]
139 [ k6*x0 ]
140 [      ]
141 [ k7*x1 ]
142 [      ]
143 [ k8*x2 ]
144 [      ]
145 [ k9*x3 ]
146
147 Differential equations functions (polynomials) vector:
148 [      2      ]
149 [-k0*x0*x2 + k2*x2 - k4*x0 + k5*x1 - k6*x0 + k8*x2]
150 [      ]
151 [      2      ]
152 [-k1*x1*x3 + k3*x3 + k4*x0 - k5*x1 - k7*x1 + k9*x3]
153 [      ]
154 [      2      ]
155 [      k0*x0*x2 - k2*x2 + k6*x0 - k8*x2      ]
156 [      ]
157 [      2      ]
158 [      k1*x1*x3 - k3*x3 + k7*x1 - k9*x3      ]
159
160 Warning: Stoichiometric Matrix is SINGULAR!
161
162
163 ===== ( - 2 - ) =====
164 *** Stoichiometric Network Analysis using the algorithm from reference [3] in README.md ***
165
166 Adding rows to the Stoichiometric Matrix encoding the dual pairs conditions in order to make
167 the cone defined by the Extreme Currents smaller and 'more precise'
168
169 Extended Stoichiometric matrix with rows indicating dual pairs
170 [[-1.  0.  1.  0. -1.  1. -1.  0.  1.  0.]
171 [ 0. -1.  0.  1.  1. -1.  0. -1.  0.  1.]
172 [ 1.  0. -1.  0.  0.  0.  1.  0. -1.  0.]
173 [ 0.  1.  0. -1.  0.  0.  0.  1.  0. -1.]
174 [ 1. -1.  0.  0.  0.  0.  0.  0.  0.  0.]
175 [ 0.  0.  1. -1.  0.  0.  0.  0.  0.  0.]
176 [ 0.  0.  0.  0.  0.  0.  1. -1.  0.  0.]
177 [ 0.  0.  0.  0.  0.  0.  0.  0.  1. -1.]]
178
179 Extreme Currents Matrix (computed from Stoichiometric Matrix)
180 [[1 0 0 1 0]

```

```

181 [1 0 0 1 0]
182 [1 0 1 0 0]
183 [1 0 1 0 0]
184 [0 1 0 0 0]
185 [0 1 0 0 0]
186 [0 0 1 0 1]
187 [0 0 1 0 1]
188 [0 0 0 1 1]
189 [0 0 0 1 1]]

```

190

191 E_omega_prim matrix

192 [j0 + j3]

193 []

194 [j0 + j3]

195 []

196 [j0 + j2]

197 []

198 [j0 + j2]

199 []

200 [j1]

201 []

202 [j1]

203 []

204 [j2 + j4]

205 []

206 [j2 + j4]

207 []

208 [j3 + j4]

209 []

210 [j3 + j4]

211

212 V(J) Matrix

213 [-j0 - j1 - j2 - j3 - j4 j1 j0 + 2*j2 + j4 0

214 [

215 [j1 -j0 - j1 - j2 - j3 - j4 0 j0 + 2*j2

216 [

217 [j0 + j2 + j3 + j4 0 -j0 - 2*j2 - j4 0

218 [

219 [0 j0 + j2 + j3 + j4 0 -j0 - 2*j2

220

221]

222]

223 + j4]

224]

225]

226]

227 - j4]

228

229

230 ---- Mineur Analysis ----

```
231 Limiting mineur analysis to mineurs of size smaller or equal to 5
232
233 There are no negative terms in the determinant of any mineur of size <= 5
234 According to this criteria the system seems to be stable.
235
236 ===== ( - 6 - ) =====
237
238 === Algorithm based on the Frank inequality, according to reference [5] in README.md file ===
239
240 *** Solving semialgebraic problem (sampling from a non-linear inequalities indirectly) using
241 Clarke factorization according to the Stoichiometric Network Analysis (SNA) ***
242
243 Enantiomers divided into L's and D's groups
244 (('L1', 'L2'), ('D1', 'D2'))
245
246 Stoichiometric Matrix is SINGULAR!
247
248 Adding rows to the Stoichiometric Matrix encoding the dual pairs conditions in order to make
249 the cone defined by the Extreme Currents smaller and 'more precise'
250
251 Extended Stoichiometric matrix with rows indicating dual pairs
252 [[-1.  0.  1.  0. -1.  1. -1.  0.  1.  0.]
253  [ 0. -1.  0.  1.  1. -1.  0. -1.  0.  1.]
254  [ 1.  0. -1.  0.  0.  0.  1.  0. -1.  0.]
255  [ 0.  1.  0. -1.  0.  0.  0.  1.  0. -1.]
256  [ 1. -1.  0.  0.  0.  0.  0.  0.  0.  0.]
257  [ 0.  0.  1. -1.  0.  0.  0.  0.  0.  0.]
258  [ 0.  0.  0.  0.  0.  0.  1. -1.  0.  0.]
259  [ 0.  0.  0.  0.  0.  0.  0.  0.  1. -1.]]
260
261 Extreme Currents Matrix (computed from Stoichiometric Matrix)
262 [[1 0 0 1 0]
263  [1 0 0 1 0]
264  [1 0 1 0 0]
265  [1 0 1 0 0]
266  [0 1 0 0 0]
267  [0 1 0 0 0]
268  [0 0 1 0 1]
269  [0 0 1 0 1]
270  [0 0 0 1 1]
271  [0 0 0 1 1]]
272
273 E_omega column
274 [j0 + j3]
275 [      ]
276 [j0 + j3]
277 [      ]
278 [j0 + j2]
279 [      ]
280 [j0 + j2]
```

```

281 [      ]
282 [ j1  ]
283 [      ]
284 [ j1  ]
285 [      ]
286 [j2 + j4]
287 [      ]
288 [j2 + j4]
289 [      ]
290 [j3 + j4]
291 [      ]
292 [j3 + j4]
293
294 V(J) Matrix
295 [-j0 - j1 - j2 - j3 - j4      j1      j0 + 2*j2 + j4      0
296 [
297 [      j1      -j0 - j1 - j2 - j3 - j4      0      j0 + 2*j2
298 [
299 [  j0 + j2 + j3 + j4      0      -j0 - 2*j2 - j4      0
300 [
301 [      0      j0 + j2 + j3 + j4      0      -j0 - 2*j2
302
303 ]
304 ]
305 + j4 ]
306 ]
307 ]
308 ]
309 - j4]
310
311 V(J) matrix reshaped to show clearly symetry [A B; B A] in Matrix
312 [-j0 - j1 - j2 - j3 - j4  j0 + 2*j2 + j4      j1      0
313 [
314 [  j0 + j2 + j3 + j4      -j0 - 2*j2 - j4      0      0
315 [
316 [      j1      0      -j0 - j1 - j2 - j3 - j4  j0 + 2*j2
317 [
318 [      0      0      j0 + j2 + j3 + j4      -j0 - 2*j2
319
320 ]
321 ]
322 ]
323 ]
324 + j4 ]
325 ]
326 - j4]
327
328 'A' Matrix
329 [-j0 - j1 - j2 - j3 - j4  j0 + 2*j2 + j4 ]
330 [      ]

```

```

331 [  j0 + j2 + j3 + j4      -j0 - 2*j2 - j4]
332
333 'B' Matrix
334 [j1  0]
335 [   ]
336 [0  0]
337
338 New V(J) Matrix (A - B)
339 [-j0 - 2*j1 - j2 - j3 - j4  j0 + 2*j2 + j4 ]
340 [                               ]
341 [  j0 + j2 + j3 + j4      -j0 - 2*j2 - j4]
342
343
344 ---- Mineur Analysis ----
345 Limiting mineur analysis to mineurs of size smaller or equal to 5
346
347 There are no negative terms in the determinant of any mineur of size <= 5
348 According to this criteria the system seems to be stable.
349 *****
350 *** Calvin - pseudoquiral ***
351 *****

352 2.2. The replicator model of Hochberg and Ribo

353 2.2.1. The input file

354 # run with:
355 # > python -m listanalchem --model models/Replicator-HR-equals.py
356
357 modelname = 'Replicator Hochberg-Ribo-Equals'
358 species = ['R1D', 'R1L', 'R2D', 'R2L', 'A']
359 reactions = [
360     'A + R1D + R2D -> 2 R1D + R2D', # 0
361     '2 R1D + R2D -> A + R1D + R2D', # 1
362     'A + R1L + R2L -> 2 R1L + R2L', # 2
363     '2 R1L + R2L -> A + R1L + R2L', # 3
364     'A + R1D + R2D -> R1D + 2 R2D', # 4
365     'R1D + 2 R2D -> A + R1D + R2D', # 5
366     'A + R1L + R2L -> R1L + 2 R2L', # 6
367     'R1L + 2 R2L -> A + R1L + R2L', # 7
368     'R1D ->', # 8
369     'R2D ->', # 9
370     'R1L ->', # 10
371     'R2L ->', # 11
372     '-> A', # 12
373     'A ->' # 13
374 ]
375 dual_pairs = [(0, 2), (1, 3), (4, 6), (5, 7), (8, 9), (9, 10), (10, 11), (11,13), (2,4), (3,5)]
376
377 analyses = {
378     "trace-determinant": {

```



```
379     "enabled": False,
380     "2by2-jacobian": True
381 },
382 "sna": {
383     "enabled": False,
384     "dual-pairs-in-ec": True,
385     # "instability-heuristic": "trace-determinant",
386     # "instability-heuristic": "characteristic-polynomial",
387     "instability-heuristic": "mineurs",
388     "sum-mineurs": True,
389     "max-mineur-search-stop": 5,
390     "simplification-tries": 1000,
391     "num-samples": 10,
392     "samples-folder": 'samples/replicator',
393 },
394 "six-categories": {
395     "enabled": False,
396     "num-samples": 10,
397     "samples-folder": 'samples/iwamoto_imperfect',
398 },
399 "frank-ineq-nonlinear": {
400     "enabled": False
401 },
402 "frank-ineq-linear": {
403     "enabled": False,
404     "dual-pairs-in-ec": True,
405     "num-samples": 1,
406     "samples-folder": None,
407     "samples-for-proportion": 1000
408 },
409 "frank-pseudoquiral": {
410     "enabled": True,
411     "enantiomeric-pairs": [(0, 1), (2, 3)],
412     "dual-pairs-in-ec": True,
413     # "instability-heuristic": "trace-determinant",
414     # "instability-heuristic": "characteristic-polynomial",
415     "instability-heuristic": "mineurs",
416     "sum-mineurs": True,
417     "max-mineur-search-stop": 5,
418     "simplification-tries": 12000,
419     "num-samples": 10,
420     "samples-folder": 'samples/replicator',
421 }
422 }
```

423 2.2.2. The output file

424 Welcome to Listanalchem!

425 Remember to check the model you're loading, it must:

426 * Have a name 'modelname'

427 * Have a list of species 'species' (note: the first two species will be considered

```

428 enantiomers, all other species do not)
429 * Have a non-empty list of reactions
430
431 *****
432 *** Replicator Hochberg-Ribo-Equals ***
433 *****
434
435 Species:
436 ['R1D', 'R1L', 'R2D', 'R2L', 'A']
437 Reactions list:
438 ['R1D + R2D + A -> 2 R1D + R2D',
439  '2 R1D + R2D -> R1D + R2D + A',
440  'R1L + R2L + A -> 2 R1L + R2L',
441  '2 R1L + R2L -> R1L + R2L + A',
442  'R1D + R2D + A -> R1D + 2 R2D',
443  'R1D + 2 R2D -> R1D + R2D + A',
444  'R1L + R2L + A -> R1L + 2 R2L',
445  'R1L + 2 R2L -> R1L + R2L + A',
446  'R1D -> ',
447  'R2D -> ',
448  'R1L -> ',
449  'R2L -> ',
450  ' -> A',
451  'A -> ']
452 Dual pair equations (equations are numbered from 0 to 13):
453 [(0, 2),
454  (1, 3),
455  (4, 6),
456  (5, 7),
457  (8, 9),
458  (9, 10),
459  (10, 11),
460  (11, 13),
461  (2, 4),
462  (3, 5)]
463 Regular reactions (not dual)
464 [12]
465
466
467 Stoichiometric Matrix:
468 [[ 1 -1 0 0 0 0 0 0 -1 0 0 0 0 0]
469  [ 0 0 1 -1 0 0 0 0 0 0 -1 0 0 0]
470  [ 0 0 0 0 1 -1 0 0 0 -1 0 0 0 0]
471  [ 0 0 0 0 0 0 1 -1 0 0 0 -1 0 0]
472  [-1 1 -1 1 -1 1 -1 1 0 0 0 0 1 -1]]
473
474 Reactions Order Matrix:
475 [[1 2 0 0 1 1 0 0 1 0 0 0 0 0]
476  [0 0 1 2 0 0 1 1 0 0 1 0 0 0]
477  [1 1 0 0 1 2 0 0 0 1 0 0 0 0]]

```

```

478 [0 0 1 1 0 0 1 2 0 0 0 1 0 0]
479 [1 0 1 0 1 0 1 0 0 0 0 0 0 1]]

```

480

481 Velocity Function:

482 [k0*x0*x2*x4]

483 []

484 [2]

485 [k1*x0 *x2]

486 []

487 [k2*x1*x3*x4]

488 []

489 [2]

490 [k3*x1 *x3]

491 []

492 [k4*x0*x2*x4]

493 []

494 [2]

495 [k5*x0*x2]

496 []

497 [k6*x1*x3*x4]

498 []

499 [2]

500 [k7*x1*x3]

501 []

502 [k8*x0]

503 []

504 [k9*x2]

505 []

506 [k10*x1]

507 []

508 [k11*x3]

509 []

510 [k12]

511 []

512 [k13*x4]

513

514 Differential equations functions (polynomials) vector:

515 [

2

516 [

k0*x0*x2*x4 - k1*x0 *x2 - k8*x0

517 [

2

519 [

-k10*x1 + k2*x1*x3*x4 - k3*x1 *x3

520 [

2

522 [

k4*x0*x2*x4 - k5*x0*x2 - k9*x2

523 [

2

525 [

-k11*x3 + k6*x1*x3*x4 - k7*x1*x3

526 [

527 [

2

2

```

528 [-k0*x0*x2*x4 + k1*x0 *x2 + k12 - k13*x4 - k2*x1*x3*x4 + k3*x1 *x3 - k4*x0*x2*
529
530 ]
531 ]
532 ]
533 ]
534 ]
535 ]
536 ]
537 ]
538 ]
539 ]
540 ]
541 ]
542      2      2]
543 x4 + k5*x0*x2 - k6*x1*x3*x4 + k7*x1*x3 ]
544
545
546 ===== ( - 6 - ) =====
547
548 === Algorithm based on the Frank inequality, according to reference [5] in README.md file ===
549
550 *** Solving semialgebraic problem (sampling from a non-linear inequalities indirectly)
551 using Clarke factorization according to the Stoichiometric Network Analysis (SNA) ***
552
553 Enantiomers divided into L's and D's groups
554 (('R1D', 'R2D'), ('R1L', 'R2L'))
555
556 Adding rows to the Stoichiometric Matrix encoding the dual pairs conditions in order to
557 make the cone defined by the Extreme Currents smaller and 'more precise'
558
559 Extended Stoichiometric matrix with rows indicating dual pairs
560 [[ 1. -1. 0. 0. 0. 0. 0. 0. -1. 0. 0. 0. 0.]
561 [ 0. 0. 1. -1. 0. 0. 0. 0. 0. 0. -1. 0. 0.]
562 [ 0. 0. 0. 0. 1. -1. 0. 0. 0. -1. 0. 0. 0.]
563 [ 0. 0. 0. 0. 0. 0. 1. -1. 0. 0. 0. -1. 0.]
564 [-1. 1. -1. 1. -1. 1. -1. 1. 0. 0. 0. 0. 1. -1.]
565 [ 1. 0. -1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
566 [ 0. 1. 0. -1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
567 [ 0. 0. 0. 0. 1. 0. -1. 0. 0. 0. 0. 0. 0. 0.]
568 [ 0. 0. 0. 0. 0. 1. 0. -1. 0. 0. 0. 0. 0. 0.]
569 [ 0. 0. 0. 0. 0. 0. 0. 0. 1. -1. 0. 0. 0. 0.]
570 [ 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. -1. 0. 0. 0.]
571 [ 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. -1. 0. 0.]
572 [ 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. -1.]
573 [ 0. 0. 1. 0. -1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
574 [ 0. 0. 0. 1. 0. -1. 0. 0. 0. 0. 0. 0. 0. 0.]]
575
576 Extreme Currents Matrix (computed from Stoichiometric Matrix)
577 [[1 1]

```

```

578 [1 0]
579 [1 1]
580 [1 0]
581 [1 1]
582 [1 0]
583 [1 1]
584 [1 0]
585 [0 1]
586 [0 1]
587 [0 1]
588 [0 1]
589 [0 5]
590 [0 1]]
591
592 E_omega column
593 [j0 + j1]
594 [      ]
595 [ j0  ]
596 [      ]
597 [j0 + j1]
598 [      ]
599 [ j0  ]
600 [      ]
601 [j0 + j1]
602 [      ]
603 [ j0  ]
604 [      ]
605 [j0 + j1]
606 [      ]
607 [ j0  ]
608 [      ]
609 [ j1  ]
610 [      ]
611 [ j1  ]
612 [      ]
613 [ j1  ]
614 [      ]
615 [ j1  ]
616 [      ]
617 [ 5*j1 ]
618 [      ]
619 [ j1  ]
620
621 V(J) Matrix
622 [ -j0      0      j1      0      j0 + j1  ]
623 [                                     ]
624 [   0     -j0      0      j1      j0 + j1  ]
625 [                                     ]
626 [  j1      0     -j0      0      j0 + j1  ]
627 [                                     ]

```

```

628 [ 0      j1      0      -j0      j0 + j1 ]
629 [
630 [j0 - 2*j1  j0 - 2*j1  j0 - 2*j1  j0 - 2*j1  -4*j0 - 5*j1]
631
632 V(J) matrix reshaped to show clearly symetry [A B; B A] in Matrix
633 [ -j0      j1      0      0      j0 + j1 ]
634 [
635 [ j1      -j0      0      0      j0 + j1 ]
636 [
637 [ 0      0      -j0      j1      j0 + j1 ]
638 [
639 [ 0      0      j1      -j0      j0 + j1 ]
640 [
641 [j0 - 2*j1  j0 - 2*j1  j0 - 2*j1  j0 - 2*j1  -4*j0 - 5*j1]
642
643 'A' Matrix
644 [-j0  j1 ]
645 [
646 [j1  -j0]
647
648 'B' Matrix
649 [0 0]
650 [ ]
651 [0 0]
652
653 New V(J) Matrix (A - B)
654 [-j0  j1 ]
655 [
656 [j1  -j0]
657
658
659 ---- Mineur Analysis ----
660 Limiting mineur analysis to mineurs of size smaller or equal to 5
661
662 Mineur:
663 Poly(j0**2 - j1**2, j0, j1, domain='ZZ')
664 Compounds number: [(0, 1)]
665
666 -----
667 Restrictions to solve
668 [j0 > 0, j1 > 0, -j0**2 + j1**2 > 0]
669
670 Trying to find a solution with the restrictions
671
672 Found a solution on try number 1 of 12000
673
674 (Randomly) Chosen variable values:
675 [j1: 0.814798351563,
676 ]
677

```

```

678 Values that variables can take given the already chosen (see above) values for variables [j1]
679 [j0 = Interval.open(0, 0.8147980000000000)
680 ]

```

681

```

682 Sampling 10 points from js intervals
683 Saving sampled models to 'samples/replicator'
684 *****
685 *** Replicator Hochberg-Ribo-Equals ***
686 *****

```

687 2.3. The APED model of Plasson et. al

688 2.3.1. The input file

```

689 # APED model 2004. Plasson-Bersini-Commeyras.
690
691 # run with "python -m listanalchem --model models/APED-2004.py"
692
693 modelname = 'APED'
694 species = ['L', 'La', 'D', 'Da', 'LL', 'DL', 'LD', 'DD']
695 reactions = [
696     'L -> La',      # 0 - ka
697     'D -> Da',      # 1 - ka
698     'La -> L',      # 2 - kb
699     'Da -> D',      # 3 - kb
700     'La + L -> LL', # 4 - kp
701     'Da + L -> DL', # 5 - kalfap
702     'La + D -> LD', # 6 - kalfap
703     'Da + D -> DD', # 7 - kp
704     'LL -> L + L',  # 8 - kh
705     'DL -> L + D',  # 9 - kbetah
706     'LD -> L + D',  # 10 - kbetah
707     'DD -> D + D',  # 11 - kh
708     'LD -> DD',     # 12 - ke
709     'DD -> LD',     # 13 - kgammae
710     'LL -> DL',     # 14 - kgammae
711     'DL -> LL'      # 15 - ke
712 ]
713
714 dual_pairs = [(0,1),(2,3),(4,7),(5,6),(8,11),(9,10),(12,15),(13,14)]
715
716 analyses = {
717     "sna": {
718         "enabled": False,
719         "dual-pairs-in-ec": True,
720         # "instability-heuristic": "trace-determinant",
721         # "instability-heuristic": "characteristic-polynomial",
722         "instability-heuristic": "mineurs",
723         "sum-mineurs": True,
724         "max-mineur-search-stop": 5,
725         "simplification-tries": 10000,

```

```

726     "num-samples": 10,
727     "samples-folder": 'samples/aped_2004',
728   },
729   "frank-pseudoquiral": {
730     "enabled": True,
731     "enantiomeric-pairs": [(0, 2), (1, 3), (4, 7), (5, 6)],
732     "dual-pairs-in-ec": True,
733     # "instability-heuristic": "trace-determinant",
734     # "instability-heuristic": "characteristic-polynomial",
735     "instability-heuristic": "mineurs",
736     "sum-mineurs": True,
737     "max-mineur-search-stop": 5,
738     "simplification-tries": 10000,
739     "num-samples": 10,
740     "samples-folder": 'samples/aped_2004',
741   }
742 }

```

743 2.3.2. The output file

```

744 Welcome to Listanalchem!
745 Remember to check the model you're loading, it must:
746 * Have a name 'modelname'
747 * Have a list of species 'species' (note: the first two species will be considered
748   enantiomers, all other species do not)
749 * Have a non-empty list of reactions
750
751 *****
752 *** APED ***
753 *****
754
755 Species:
756 ['L', 'La', 'D', 'Da', 'LL', 'DL', 'LD', 'DD']
757 Reactions list:
758 ['L -> La',
759  'D -> Da',
760  'La -> L',
761  'Da -> D',
762  'L + La -> LL',
763  'L + Da -> DL',
764  'La + D -> LD',
765  'D + Da -> DD',
766  'LL -> 2 L',
767  'DL -> L + D',
768  'LD -> L + D',
769  'DD -> 2 D',
770  'LD -> DD',
771  'DD -> LD',
772  'LL -> DL',
773  'DL -> LL']
774 Dual pair equations (equations are numbered from 0 to 15):

```


775 [(0, 1), (2, 3), (4, 7), (5, 6), (8, 11), (9, 10), (12, 15), (13, 14)]

776

777 Stoichiometric Matrix:

```
778 [[-1  0  1  0 -1 -1  0  0  2  1  1  0  0  0  0  0]
779 [ 1  0 -1  0 -1  0 -1  0  0  0  0  0  0  0  0  0]
780 [ 0 -1  0  1  0  0 -1 -1  0  1  1  2  0  0  0  0]
781 [ 0  1  0 -1  0 -1  0 -1  0  0  0  0  0  0  0  0]
782 [ 0  0  0  0  1  0  0  0 -1  0  0  0  0  0 -1  1]
783 [ 0  0  0  0  0  1  0  0  0 -1  0  0  0  0  1 -1]
784 [ 0  0  0  0  0  0  1  0  0  0 -1  0 -1  1  0  0]
785 [ 0  0  0  0  0  0  0  1  0  0  0 -1  1 -1  0  0]]
```

786

787 Reactions Order Matrix:

```
788 [[1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0]
789 [0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0]
790 [0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0]
791 [0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0]
792 [0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0]
793 [0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1]
794 [0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0]
795 [0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0]]
```

796

797 Velocity Function:

```
798 [ k0*x0 ]
799 [      ]
800 [ k1*x2 ]
801 [      ]
802 [ k2*x1 ]
803 [      ]
804 [ k3*x3 ]
805 [      ]
806 [k4*x0*x1]
807 [      ]
808 [k5*x0*x3]
809 [      ]
810 [k6*x1*x2]
811 [      ]
812 [k7*x2*x3]
813 [      ]
814 [ k8*x4 ]
815 [      ]
816 [ k9*x5 ]
817 [      ]
818 [ k10*x6 ]
819 [      ]
820 [ k11*x7 ]
821 [      ]
822 [ k12*x6 ]
823 [      ]
824 [ k13*x7 ]
```

```

825 [          ]
826 [ k14*x4 ]
827 [          ]
828 [ k15*x5 ]
829
830 Differential equations functions (polynomials) vector:
831 [-k0*x0 + k10*x6 + k2*x1 - k4*x0*x1 - k5*x0*x3 + 2*k8*x4 + k9*x5 ]
832 [          ]
833 [          k0*x0 - k2*x1 - k4*x0*x1 - k6*x1*x2          ]
834 [          ]
835 [-k1*x2 + k10*x6 + 2*k11*x7 + k3*x3 - k6*x1*x2 - k7*x2*x3 + k9*x5]
836 [          ]
837 [          k1*x2 - k3*x3 - k5*x0*x3 - k7*x2*x3          ]
838 [          ]
839 [          -k14*x4 + k15*x5 + k4*x0*x1 - k8*x4          ]
840 [          ]
841 [          k14*x4 - k15*x5 + k5*x0*x3 - k9*x5          ]
842 [          ]
843 [          -k10*x6 - k12*x6 + k13*x7 + k6*x1*x2          ]
844 [          ]
845 [          -k11*x7 + k12*x6 - k13*x7 + k7*x2*x3          ]
846
847 Warning: Stoichiometric Matrix is SINGULAR!
848
849
850 ===== ( - 6 - ) =====
851
852 === Algorithm based on the Frank inequality, according to reference [5] in README.md file ===
853
854 *** Solving semialgebraic problem (sampling from a non-linear inequalities indirectly)
855 using Clarke factorization according to the Stoichiometric Network Analysis (SNA) ***
856
857 Enantiomers divided into L's and D's groups
858 (('L', 'La', 'LL', 'DL'), ('D', 'Da', 'DD', 'LD'))
859
860 Stoichiometric Matrix is SINGULAR!
861
862 Adding rows to the Stoichiometric Matrix encoding the dual pairs conditions in order to
863 make the cone defined by the Extreme Currents smaller and 'more precise'
864
865 Extended Stoichiometric matrix with rows indicating dual pairs
866 [[-1.  0.  1.  0. -1. -1.  0.  0.  2.  1.  1.  0.  0.  0.  0.]
867 [ 1.  0. -1.  0. -1.  0. -1.  0.  0.  0.  0.  0.  0.  0.  0.]
868 [ 0. -1.  0.  1.  0.  0. -1. -1.  0.  1.  1.  2.  0.  0.  0.]
869 [ 0.  1.  0. -1.  0. -1.  0. -1.  0.  0.  0.  0.  0.  0.  0.]
870 [ 0.  0.  0.  0.  1.  0.  0.  0. -1.  0.  0.  0.  0.  0. -1.  1.]
871 [ 0.  0.  0.  0.  0.  1.  0.  0.  0. -1.  0.  0.  0.  0.  1. -1.]
872 [ 0.  0.  0.  0.  0.  0.  1.  0.  0.  0. -1.  0. -1.  1.  0.  0.]
873 [ 0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0. -1.  1. -1.  0.  0.]
874 [ 1. -1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]

```

```

875 [ 0.  0.  1. -1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
876 [ 0.  0.  0.  0.  1.  0.  0. -1.  0.  0.  0.  0.  0.  0.  0.]
877 [ 0.  0.  0.  0.  0.  1. -1.  0.  0.  0.  0.  0.  0.  0.  0.]
878 [ 0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0. -1.  0.  0.  0.]
879 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  1. -1.  0.  0.  0.  0.]
880 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0. -1.]
881 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1. -1.  0.]]

```

882

883 Extreme Currents Matrix (computed from Stoichiometric Matrix)

```

884 [[1 1 1 1 1 0]
885 [1 1 1 1 1 0]
886 [1 0 0 0 0 0]
887 [1 0 0 0 0 0]
888 [0 0 1 1 0 0]
889 [0 1 0 0 1 0]
890 [0 1 0 0 1 0]
891 [0 0 1 1 0 0]
892 [0 0 1 0 1 0]
893 [0 1 0 1 0 0]
894 [0 1 0 1 0 0]
895 [0 0 1 0 1 0]
896 [0 0 0 0 1 1]
897 [0 0 0 1 0 1]
898 [0 0 0 1 0 1]
899 [0 0 0 0 1 1]]

```

900

901 E_omega column

```

902 [j0 + j1 + j2 + j3 + j4]
903 [
904 [j0 + j1 + j2 + j3 + j4]
905 [
906 [ j0 ]
907 [
908 [ j0 ]
909 [
910 [ j2 + j3 ]
911 [
912 [ j1 + j4 ]
913 [
914 [ j1 + j4 ]
915 [
916 [ j2 + j3 ]
917 [
918 [ j2 + j4 ]
919 [
920 [ j1 + j3 ]
921 [
922 [ j1 + j3 ]
923 [
924 [ j2 + j4 ]

```

```

925 [
926 [      j4 + j5      ]
927 [                  ]
928 [      j3 + j5      ]
929 [                  ]
930 [      j3 + j5      ]
931 [                  ]
932 [      j4 + j5      ]
933
934 V(J) Matrix
935 [-j0 - 2*j1 - 2*j2 - 2*j3 - 2*j4      j0 - j2 - j3      0
936 [
937 [      j0 + j1 + j4      -j0 - j1 - j2 - j3 - j4      -j1 - j4
938 [
939 [      0      -j1 - j4      -j0 - 2*j1 - 2*j2 -
940 [
941 [      -j1 - j4      0      j0 + j1 +
942 [
943 [      j2 + j3      j2 + j3      0
944 [
945 [      j1 + j4      0      0
946 [
947 [      0      j1 + j4      j1 + j4
948 [
949 [      0      0      j2 + j3
950
951 [      -j1 - j4      2*j2 + 2*j4      j1 + j3
952
953 [      0      0      0
954
955 [ 2*j3 - 2*j4      j0 - j2 - j3      0      j1 + j3
956
957 [ j4      -j0 - j1 - j2 - j3 - j4      0      0
958
959 [      0      -j2 - j3 - j4 - j5      j4 + j5
960
961 [      j1 + j4      j3 + j5      -j1 - j3 - j4 - j5
962
963 [      0      0      0
964
965 [      j2 + j3      0      0
966
967 [      j1 + j3      0      ]
968 [                  ]
969 [      0      0      ]
970 [                  ]
971 [      j1 + j3      2*j2 + 2*j4      ]
972 [                  ]
973 [      0      0      ]
974 [                  ]

```

```

975         0           0           ]
976         ]
977         0           0           ]
978         ]
979     -j1 - j3 - j4 - j5       j3 + j5       ]
980         ]
981         j4 + j5       -j2 - j3 - j4 - j5]
982
983 V(J) matrix reshaped to show clearly symetry [A B; B A] in Matrix
984 [-j0 - 2*j1 - 2*j2 - 2*j3 - 2*j4       j0 - j2 - j3       2*j2 + 2*j4
985 [
986 [       j0 + j1 + j4       -j0 - j1 - j2 - j3 - j4       0
987 [
988 [       j2 + j3       j2 + j3       -j2 - j3 - j4 - j5
989 [
990 [       j1 + j4       0       j3 + j5
991 [
992 [       0       -j1 - j4       0
993 [
994 [       -j1 - j4       0       0
995 [
996 [       0       0       0
997 [
998 [       0       j1 + j4       0
999
1000     j1 + j3       0       -j1 - j4
1001
1002     0       -j1 - j4       0
1003
1004     j4 + j5       0       0
1005
1006     -j1 - j3 - j4 - j5       0       j1 + j4
1007
1008     j1 + j3       -j0 - 2*j1 - 2*j2 - 2*j3 - 2*j4       j0 - j2 - j3
1009
1010     0       j0 + j1 + j4       -j0 - j1 - j2 - j3 - j4
1011
1012     0       j2 + j3       j2 + j3
1013
1014     0       j1 + j4       0
1015
1016     0       j1 + j3       ]
1017         ]
1018     0       0       ]
1019         ]
1020     0       0       ]
1021         ]
1022     0       0       ]
1023         ]
1024     2*j2 + 2*j4       j1 + j3       ]

```

```

1025         ]
1026         0             0             ]
1027         ]
1028     -j2 - j3 - j4 - j5     j4 + j5     ]
1029         ]
1030         j3 + j5     -j1 - j3 - j4 - j5]
1031
1032 'A' Matrix
1033 [-j0 - 2*j1 - 2*j2 - 2*j3 - 2*j4     j0 - j2 - j3     2*j2 + 2*j4
1034 [
1035 [     j0 + j1 + j4     -j0 - j1 - j2 - j3 - j4     0
1036 [
1037 [     j2 + j3     j2 + j3     -j2 - j3 - j4 - j5
1038 [
1039 [     j1 + j4     0     j3 + j5
1040
1041     j1 + j3     ]
1042     ]
1043     0     ]
1044     ]
1045     j4 + j5     ]
1046     ]
1047     -j1 - j3 - j4 - j5]
1048
1049 'B' Matrix
1050 [  0     -j1 - j4  0  j1 + j3]
1051 [     ]
1052 [-j1 - j4     0     0     0 ]
1053 [     ]
1054 [  0     0     0     0 ]
1055 [     ]
1056 [  0     j1 + j4  0     0 ]
1057
1058 New V(J) Matrix (A - B)
1059 [-j0 - 2*j1 - 2*j2 - 2*j3 - 2*j4  j0 + j1 - j2 - j3 + j4     2*j2 + 2*j4
1060 [
1061 [     j0 + 2*j1 + 2*j4     -j0 - j1 - j2 - j3 - j4     0
1062 [
1063 [     j2 + j3     j2 + j3     -j2 - j3 - j4 - j5
1064 [
1065 [     j1 + j4     -j1 - j4     j3 + j5
1066
1067     0     ]
1068     ]
1069     0     ]
1070     ]
1071     j4 + j5     ]
1072     ]
1073     -j1 - j3 - j4 - j5]
1074

```

```

1075
1076 ---- Mineur Analysis ----
1077 Limiting mineur analysis to mineurs of size smaller or equal to 5
1078
1079 Mineur:
1080 Poly(4*j0*j1*j2*j3 + 4*j0*j1*j2*j5 + 4*j0*j1*j3**2 + 4*j0*j1*j3*j5 + 4*j0*j2*j
1081 3**2 + 4*j0*j2*j3*j5 + 4*j0*j3**3 + 4*j0*j3**2*j5 + 6*j1**2*j2*j3 + 2*j1**2*j2
1082 *j4 + 8*j1**2*j2*j5 + 6*j1**2*j3**2 + 6*j1**2*j3*j5 + 2*j1**2*j4**2 + 2*j1**2*
1083 j4*j5 + 2*j1*j2**2*j3 - 2*j1*j2**2*j4 + 10*j1*j2*j3**2 + 4*j1*j2*j3*j4 + 8*j1*
1084 j2*j3*j5 + 2*j1*j2*j4**2 + 8*j1*j2*j4*j5 + 8*j1*j3**3 + 6*j1*j3**2*j4 + 8*j1*j
1085 3**2*j5 - 2*j1*j3*j4**2 + 4*j1*j3*j4*j5 + 4*j1*j4**3 + 4*j1*j4**2*j5 + 2*j2**2
1086 *j3**2 + 2*j2**2*j3*j5 - 2*j2**2*j4**2 - 2*j2**2*j4*j5 + 4*j2*j3**3 + 6*j2*j3*
1087 *2*j4 + 4*j2*j3**2*j5 - 2*j2*j3*j4**2 + 4*j2*j3*j4*j5 + 2*j3**4 + 6*j3**3*j4 +
1088 2*j3**3*j5 + 6*j3**2*j4*j5 - 2*j3*j4**3 - 2*j3*j4**2*j5 + 2*j4**4 + 2*j4**3*j
1089 5, j0, j1, j2, j3, j4, j5, domain='ZZ')
1090 Compounds number: [(0, 1, 2, 3)]
1091
1092 -----
1093 Restrictions to solve
1094 [j0 > 0,
1095  j1 > 0,
1096  j2 > 0,
1097  j3 > 0,
1098  j4 > 0,
1099  j5 > 0,
1100 -4*j0*j1*j2*j3 - 4*j0*j1*j2*j5 - 4*j0*j1*j3**2 - 4*j0*j1*j3*j5 -
1101 4*j0*j2*j3**2 - 4*j0*j2*j3*j5 - 4*j0*j3**3 - 4*j0*j3**2*j5 -
1102 6*j1**2*j2*j3 - 2*j1**2*j2*j4 - 8*j1**2*j2*j5 - 6*j1**2*j3**2 -
1103 6*j1**2*j3*j5 - 2*j1**2*j4**2 - 2*j1**2*j4*j5 - 2*j1*j2**2*j3 +
1104 2*j1*j2**2*j4 - 10*j1*j2*j3**2 - 4*j1*j2*j3*j4 - 8*j1*j2*j3*j5 -
1105 2*j1*j2*j4**2 - 8*j1*j2*j4*j5 - 8*j1*j3**3 - 6*j1*j3**2*j4 -
1106 8*j1*j3**2*j5 + 2*j1*j3*j4**2 - 4*j1*j3*j4*j5 - 4*j1*j4**3 -
1107 4*j1*j4**2*j5 - 2*j2**2*j3**2 - 2*j2**2*j3*j5 + 2*j2**2*j4**2 +
1108 2*j2**2*j4*j5 - 4*j2*j3**3 - 6*j2*j3**2*j4 - 4*j2*j3**2*j5 +
1109 2*j2*j3*j4**2 - 4*j2*j3*j4*j5 - 2*j3**4 - 6*j3**3*j4 - 2*j3**3*j5 -
1110 6*j3**2*j4*j5 + 2*j3*j4**3 + 2*j3*j4**2*j5 - 2*j4**4 - 2*j4**3*j5 > 0]
1111
1112 Trying to find a solution with the restrictions
1113
1114 Found a solution on try number 33 of 10000
1115
1116 (Randomly) Chosen variable values:
1117 [j3: 0.101800290882,
1118  j0: 0.529336072892,
1119  j2: 1.92129144847,
1120  j4: 1.51730047593,
1121  j1: 0.382895906423,
1122  ]
1123
1124 Values that variables can take given the already chosen (see above) values for

```

```
1125 variables [j3, j4, j1, j2, j0]
1126 [j5 = (0, 683403118501363/15617688553613300)]
1127
1128 Sampling 10 points from js intervals
1129 Saving sampled models to 'samples/aped_2004'
1130 *****
1131 *** APED ***
1132 *****

1133 2.4. The Iwamoto model

1134 2.4.1. The Iwamoto model under perfect conditions

1135 The input file

1136 # run with "python -m listanalchem --model models/Iwamoto-Perfect.py"
1137 modelname = 'Iwamoto-Perfect'
1138 species = ["L", "D", "E-L", "E-D", "A"]
1139 reactions = [
1140     "<-> A",          # 0 & 1.
1141     "A + L <-> 2L",   # 2 & 3.
1142     "A + D <-> 2D",   # 4 & 5.
1143     "L + E-L <->",    # 6 & 7.
1144     "D + E-D <->",    # 8 & 9.
1145     "-> E-L",         # 10.
1146     "-> E-D"         # 11.
1147 ]
1148 dual_pairs = [(2, 4), (3, 5), (6, 8), (7, 9), (10, 11)]
1149
1150 analyses = {
1151     "frank-pseudoquiral": {
1152         "enabled": True,
1153         "enantiomeric-pairs": [(0, 1), (2, 3)],
1154         "dual-pairs-in-ec": True,
1155         # "instability-heuristic": "trace-determinant",
1156         # "instability-heuristic": "characteristic-polynomial",
1157         "instability-heuristic": "mineurs",
1158         "sum-mineurs": True,
1159         "max-mineur-search-stop": 5,
1160         "simplification-tries": 1000,
1161         "num-samples": 10,
1162         "samples-folder": 'samples/iwamoto_perfect',
1163     }
1164 }

1165 The output file

1166 Welcome to Listanalchem!
1167 Remember to check the model you're loading, it must:
1168 * Have a name 'modelname'
1169 * Have a list of species 'species' (note: the first two species will be considered
1170   enantiomers, all other species do not)
```



```

1171 * Have a non-empty list of reactions
1172
1173 *****
1174 *** Iwamoto-Perfect ***
1175 *****
1176
1177 Species:
1178 ['L', 'D', 'E-L', 'E-D', 'A']
1179 Reactions list:
1180 [' -> A',
1181  'A -> ',
1182  'L + A -> 2 L',
1183  '2 L -> L + A',
1184  'D + A -> 2 D',
1185  '2 D -> D + A',
1186  'L + E-L -> ',
1187  ' -> L + E-L',
1188  'D + E-D -> ',
1189  ' -> D + E-D',
1190  ' -> E-L',
1191  ' -> E-D']
1192 Dual pair equations (equations are numbered from 0 to 11):
1193 [(2, 4), (3, 5), (6, 8), (7, 9), (10, 11)]
1194 Regular reactions (not dual)
1195 [0, 1]
1196
1197
1198 Stoichiometric Matrix:
1199 [[ 0  0  1 -1  0  0 -1  1  0  0  0  0]
1200  [ 0  0  0  0  1 -1  0  0 -1  1  0  0]
1201  [ 0  0  0  0  0  0 -1  1  0  0  1  0]
1202  [ 0  0  0  0  0  0  0  0 -1  1  0  1]
1203  [ 1 -1 -1  1 -1  1  0  0  0  0  0  0]]
1204
1205 Reactions Order Matrix:
1206 [[0 0 1 2 0 0 1 0 0 0 0 0]
1207  [0 0 0 0 1 2 0 0 1 0 0 0]
1208  [0 0 0 0 0 0 1 0 0 0 0 0]
1209  [0 0 0 0 0 0 0 0 1 0 0 0]
1210  [0 1 1 0 1 0 0 0 0 0 0 0]]
1211
1212 Velocity Function:
1213 [  k0  ]
1214 [      ]
1215 [ k1*x4 ]
1216 [      ]
1217 [k2*x0*x4]
1218 [      ]
1219 [      2 ]
1220 [ k3*x0 ]

```

```

1221 [      ]
1222 [k4*x1*x4]
1223 [      ]
1224 [      2 ]
1225 [ k5*x1 ]
1226 [      ]
1227 [k6*x0*x2]
1228 [      ]
1229 [  k7 ]
1230 [      ]
1231 [k8*x1*x3]
1232 [      ]
1233 [  k9 ]
1234 [      ]
1235 [ k10 ]
1236 [      ]
1237 [  k11 ]
1238
1239 Differential equations functions (polynomials) vector:
1240 [      2      ]
1241 [      k2*x0*x4 - k3*x0  - k6*x0*x2 + k7      ]
1242 [      ]
1243 [      2      ]
1244 [      k4*x1*x4 - k5*x1  - k8*x1*x3 + k9      ]
1245 [      ]
1246 [      k10 - k6*x0*x2 + k7      ]
1247 [      ]
1248 [      k11 - k8*x1*x3 + k9      ]
1249 [      ]
1250 [      2      2      ]
1251 [k0 - k1*x4 - k2*x0*x4 + k3*x0  - k4*x1*x4 + k5*x1 ]
1252
1253
1254 ===== ( - 6 - ) =====
1255
1256 === Algorithm based on the Frank inequality, according to reference [5] in README.md file ===
1257
1258 *** Solving semialgebraic problem (sampling from a non-linear inequalities indirectly) using
1259     Clarke factorization according to the Stoichiometric Network Analysis (SNA) ***
1260
1261 Enantiomers divided into L's and D's groups
1262 (('L', 'E-L'), ('D', 'E-D'))
1263
1264 Adding rows to the Stoichiometric Matrix encoding the dual pairs conditions in order to make
1265 the cone defined by the Extreme Currents smaller and 'more precise'
1266
1267 Extended Stoichiometric matrix with rows indicating dual pairs
1268 [[ 0.  0.  1. -1.  0.  0. -1.  1.  0.  0.  0.  0.]
1269 [ 0.  0.  0.  0.  1. -1.  0.  0. -1.  1.  0.  0.]
1270 [ 0.  0.  0.  0.  0.  0. -1.  1.  0.  0.  1.  0.]

```

```

1271 [ 0.  0.  0.  0.  0.  0.  0.  0.  -1.  1.  0.  1.]
1272 [ 1. -1. -1.  1. -1.  1.  0.  0.  0.  0.  0.  0.]
1273 [ 0.  0.  1.  0. -1.  0.  0.  0.  0.  0.  0.  0.]
1274 [ 0.  0.  0.  1.  0. -1.  0.  0.  0.  0.  0.  0.]
1275 [ 0.  0.  0.  0.  0.  0.  1.  0. -1.  0.  0.  0.]
1276 [ 0.  0.  0.  0.  0.  0.  0.  1.  0. -1.  0.  0.]
1277 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1. -1.]]

```

1278

1279 Extreme Currents Matrix (computed from Stoichiometric Matrix)

```

1280 [[1 0 0 2]
1281 [1 0 0 0]
1282 [0 1 0 1]
1283 [0 1 0 0]
1284 [0 1 0 1]
1285 [0 1 0 0]
1286 [0 0 1 1]
1287 [0 0 1 0]
1288 [0 0 1 1]
1289 [0 0 1 0]
1290 [0 0 0 1]
1291 [0 0 0 1]]

```

1292

1293 E_omega column

```

1294 [j0 + 2*j3]
1295 [      ]
1296 [  j0   ]
1297 [      ]
1298 [ j1 + j3 ]
1299 [      ]
1300 [  j1   ]
1301 [      ]
1302 [ j1 + j3 ]
1303 [      ]
1304 [  j1   ]
1305 [      ]
1306 [ j2 + j3 ]
1307 [      ]
1308 [  j2   ]
1309 [      ]
1310 [ j2 + j3 ]
1311 [      ]
1312 [  j2   ]
1313 [      ]
1314 [  j3   ]
1315 [      ]
1316 [  j3   ]

```

1317

1318 V(J) Matrix

```

1319 [-j1 - j2    0    -j2 - j3    0    j1 + j3    ]
1320 [                                     ]

```

```

1321 [  0      -j1 - j2      0      -j2 - j3      j1 + j3      ]
1322 [
1323 [-j2 - j3      0      -j2 - j3      0      0      ]
1324 [
1325 [  0      -j2 - j3      0      -j2 - j3      0      ]
1326 [
1327 [j1 - j3      j1 - j3      0      0      -j0 - 2*j1 - 2*j3]
1328
1329 V(J) matrix reshaped to show clearly symmetry [A B; B A] in Matrix
1330 [-j1 - j2  -j2 - j3      0      0      j1 + j3      ]
1331 [
1332 [-j2 - j3  -j2 - j3      0      0      0      ]
1333 [
1334 [  0      0      -j1 - j2  -j2 - j3      j1 + j3      ]
1335 [
1336 [  0      0      -j2 - j3  -j2 - j3      0      ]
1337 [
1338 [j1 - j3      0      j1 - j3      0      -j0 - 2*j1 - 2*j3]
1339
1340 'A' Matrix
1341 [-j1 - j2  -j2 - j3]
1342 [
1343 [-j2 - j3  -j2 - j3]
1344
1345 'B' Matrix
1346 [0 0]
1347 [
1348 [0 0]
1349
1350 New V(J) Matrix (A - B)
1351 [-j1 - j2  -j2 - j3]
1352 [
1353 [-j2 - j3  -j2 - j3]
1354
1355
1356 ---- Mineur Analysis ----
1357 Limiting mineur analysis to mineurs of size smaller or equal to 5
1358
1359 Mineur:
1360 Poly(j1*j2 + j1*j3 - j2*j3 - j3**2, j1, j2, j3, domain='ZZ')
1361 Compounds number: [(0, 1)]
1362
1363 -----
1364 Restrictions to solve
1365 [j0 > 0, j1 > 0, j2 > 0, j3 > 0, -j1*j2 - j1*j3 + j2*j3 + j3**2 > 0]
1366
1367 Trying to find a solution with the restrictions
1368
1369 Found a solution on try number 1 of 1000
1370

```

```

1371 (Randomly) Chosen variable values:
1372 [j3: 1.68144013191,
1373   j2: 0.876669794652,
1374   ]
1375
1376 Values that variables can take given the already chosen (see above) values for variables
1377 [j3, j2]
1378 [j1 = (0, 430130869234109/255810992655764)]
1379
1380 Sampling 10 points from js intervals
1381 Saving sampled models to 'samples/iwamoto_perfect'
1382 *****
1383 *** Iwamoto-Perfect ***
1384 *****

```

1385 The Iwamoto model under imperfect conditions

1386 The input file

```

1387 # run with: "python -m listanalchem --model models/Iwamoto-Imperfect.py"
1388 modelname = 'Iwamoto-Imperfect'
1389 species = ["L", "D", "E-L", "E-D", "A"]
1390 reactions = [
1391     " <-> A",          # 0 & 1.
1392     "A + L <-> 2L",    # 2 & 3.
1393     "A + L <-> L + D", # 4 & 5.
1394     "A + D <-> 2D",    # 6 & 7.
1395     "A + D <-> D + L", # 8 & 9.
1396     "L + E-L <-> ",    # 10 & 11.
1397     "D + E-D <-> ",    # 12 & 13.
1398     "D + E-L <-> ",    # 14 & 15.
1399     "L + E-D <-> ",    # 16 & 17.
1400     " -> E-L",        # 18.
1401     " -> E-D"         # 19.
1402 ]
1403 dual_pairs = [(2,6),(4,8),(3,7),(5,9),(10,16),(12,14),(11,17),(13,15),(18,19)]
1404
1405 analyses = {
1406     "sna": {
1407         "enabled": False,
1408         "dual-pairs-in-ec": True,
1409         # "instability-heuristic": "trace-determinant",
1410         # "instability-heuristic": "characteristic-polynomial",
1411         "instability-heuristic": "mineurs",
1412         "sum-mineurs": True,
1413         "max-mineur-search-stop": 5,
1414         "simplification-tries": 10000,
1415         "num-samples": 10,
1416         "samples-folder": 'samples/iwamoto-imperfect',
1417     },
1418     "frank-pseudoquiral": {

```

```

1419     "enabled": True,
1420     "enantiomeric-pairs": [(0, 1), (2, 3)],
1421     "dual-pairs-in-ec": True,
1422     # "instability-heuristic": "trace-determinant",
1423     # "instability-heuristic": "characteristic-polynomial",
1424     "instability-heuristic": "mineurs",
1425     "sum-mineurs": True,
1426     "max-mineur-search-stop": 5,
1427     "simplification-tries": 10000,
1428     "num-samples": 10,
1429     "samples-folder": 'samples/iwamoto-imperfect',
1430 }
1431 }

```

1432 The output file

```

1433 Welcome to Listanalchem!
1434 Remember to check the model you're loading, it must:
1435 * Have a name 'modelname'
1436 * Have a list of species 'species' (note: the first two species will be considered
1437   enantiomers, all other species do not)
1438 * Have a non-empty list of reactions
1439
1440 *****
1441 *** Iwamoto-Imperfect ***
1442 *****
1443
1444 Species:
1445 ['L', 'D', 'E-L', 'E-D', 'A']
1446 Reactions list:
1447 [' -> A',
1448  'A -> ',
1449  'L + A -> 2 L',
1450  '2 L -> L + A',
1451  'L + A -> L + D',
1452  'L + D -> L + A',
1453  'D + A -> 2 D',
1454  '2 D -> D + A',
1455  'D + A -> L + D',
1456  'L + D -> D + A',
1457  'L + E-L -> ',
1458  ' -> L + E-L',
1459  'D + E-D -> ',
1460  ' -> D + E-D',
1461  'D + E-L -> ',
1462  ' -> D + E-L',
1463  'L + E-D -> ',
1464  ' -> L + E-D',
1465  ' -> E-L',
1466  ' -> E-D']
1467 Dual pair equations (equations are numbered from 0 to 19):

```

```

1468 [(2, 6),
1469 (4, 8),
1470 (3, 7),
1471 (5, 9),
1472 (10, 16),
1473 (12, 14),
1474 (11, 17),
1475 (13, 15),
1476 (18, 19)]
1477 Regular reactions (not dual)
1478 [0, 1]
1479
1480
1481 Stoichiometric Matrix:
1482 [[ 0  0  1 -1  0  0  0  0  1 -1 -1  1  0  0  0  0 -1  1  0  0]
1483 [ 0  0  0  0  1 -1  1 -1  0  0  0  0  0 -1  1 -1  1  0  0  0]
1484 [ 0  0  0  0  0  0  0  0  0  0  0 -1  1  0  0 -1  1  0  0  1]
1485 [ 0  0  0  0  0  0  0  0  0  0  0  0 -1  1  0  0 -1  1  0  1]
1486 [ 1 -1 -1  1 -1  1 -1  1 -1  1  0  0  0  0  0  0  0  0  0  0]]
1487
1488 Reactions Order Matrix:
1489 [[0 0 1 2 1 1 0 0 0 1 1 0 0 0 0 0 1 0 0 0]
1490 [0 0 0 0 0 1 1 2 1 1 0 0 1 0 1 0 0 0 0 0]
1491 [0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0]
1492 [0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0]
1493 [0 1 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0]]
1494
1495 Velocity Function:
1496 [ k0 ]
1497 [ ]
1498 [ k1*x4 ]
1499 [ ]
1500 [k2*x0*x4 ]
1501 [ ]
1502 [ 2 ]
1503 [ k3*x0 ]
1504 [ ]
1505 [k4*x0*x4 ]
1506 [ ]
1507 [k5*x0*x1 ]
1508 [ ]
1509 [k6*x1*x4 ]
1510 [ ]
1511 [ 2 ]
1512 [ k7*x1 ]
1513 [ ]
1514 [k8*x1*x4 ]
1515 [ ]
1516 [k9*x0*x1 ]
1517 [ ]

```

```

1518 [k10*x0*x2]
1519 [      ]
1520 [  k11  ]
1521 [      ]
1522 [k12*x1*x3]
1523 [      ]
1524 [  k13  ]
1525 [      ]
1526 [k14*x1*x2]
1527 [      ]
1528 [  k15  ]
1529 [      ]
1530 [k16*x0*x3]
1531 [      ]
1532 [  k17  ]
1533 [      ]
1534 [  k18  ]
1535 [      ]
1536 [  k19  ]
1537
1538 Differential equations functions (polynomials) vector:
1539 [
1540 [          -k10*x0*x2 + k11 - k16*x0*x3 + k17 + k2*x0*x4 - k3*x02 + k8*x1*x4 -
1541 [
1542 [
1543 [          -k12*x1*x3 + k13 - k14*x1*x2 + k15 + k4*x0*x4 - k5*x0*x1 + k6*x1*x4
1544 [
1545 [                      -k10*x0*x2 + k11 - k14*x1*x2 + k15 + k18
1546 [
1547 [                      -k12*x1*x3 + k13 - k16*x0*x3 + k17 + k19
1548 [
1549 [                      2                      2
1550 [k0 - k1*x4 - k2*x0*x4 + k3*x02 - k4*x0*x4 + k5*x0*x1 - k6*x1*x4 + k7*x12 - k8
1551
1552 [
1553 k9*x0*x1 ]
1554 [
1555 2 ]
1556 - k7*x1 ]
1557 [
1558 [
1559 [
1560 [
1561 [
1562 [
1563 *x1*x4 + k9*x0*x1]
1564
1565
1566 ===== ( - 6 - ) =====
1567

```



```
1568 === Algorithm based on the Frank inequality, according to reference [5] in README.md file ===
1569
1570 *** Solving semialgebraic problem (sampling from a non-linear inequalities indirectly) using
1571 Clarke factorization according to the Stoichiometric Network Analysis (SNA) ***
1572
1573 Enantiomers divided into L's and D's groups
1574 (('L', 'E-L'), ('D', 'E-D'))
1575
1576 Adding rows to the Stoichiometric Matrix encoding the dual pairs conditions in order to make
1577 the cone defined by the Extreme Currents smaller and 'more precise'
1578
1579 Extended Stoichiometric matrix with rows indicating dual pairs
1580 [[ 0.  0.  1. -1.  0.  0.  0.  0.  1. -1. -1.  1.  0.  0.  0.  0. -1.  1.
1581    0.  0.]
1582 [ 0.  0.  0.  0.  1. -1.  1. -1.  0.  0.  0.  0. -1.  1. -1.  1.  0.  0.
1583    0.  0.]
1584 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0. -1.  1.  0.  0. -1.  1.  0.  0.
1585    1.  0.]
1586 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0. -1.  1.  0.  0. -1.  1.
1587    0.  1.]
1588 [ 1. -1. -1.  1. -1.  1. -1.  1. -1.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.
1589    0.  0.]
1590 [ 0.  0.  1.  0.  0.  0. -1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
1591    0.  0.]
1592 [ 0.  0.  0.  0.  1.  0.  0.  0. -1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
1593    0.  0.]
1594 [ 0.  0.  0.  1.  0.  0.  0. -1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
1595    0.  0.]
1596 [ 0.  0.  0.  0.  0.  1.  0.  0.  0. -1.  0.  0.  0.  0.  0.  0.  0.  0.  0.
1597    0.  0.]
1598 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0. -1.  0.
1599    0.  0.]
1600 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0. -1.  0.  0.  0.
1601    0.  0.]
1602 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0. -1.
1603    0.  0.]
1604 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  0. -1.  0.  0.
1605    0.  0.]
1606 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
1607    1. -1.]]
1608
1609 Extreme Currents Matrix (computed from Stoichiometric Matrix)
1610 [[1 0 0 0 0 0 0 4 4]
1611 [1 0 0 0 0 0 0 0 0]
1612 [0 1 0 1 0 0 0 2 0]
1613 [0 1 1 0 0 0 0 0 0]
1614 [0 0 1 0 1 0 0 0 2]
1615 [0 0 0 1 1 0 0 0 0]
1616 [0 1 0 1 0 0 0 2 0]
1617 [0 1 1 0 0 0 0 0 0]
```

```

1618 [0 0 1 0 1 0 0 0 2]
1619 [0 0 0 1 1 0 0 0 0]
1620 [0 0 0 0 0 0 1 1 1]
1621 [0 0 0 0 0 0 1 0 0]
1622 [0 0 0 0 0 1 0 1 1]
1623 [0 0 0 0 0 1 0 0 0]
1624 [0 0 0 0 0 1 0 1 1]
1625 [0 0 0 0 0 1 0 0 0]
1626 [0 0 0 0 0 0 1 1 1]
1627 [0 0 0 0 0 0 1 0 0]
1628 [0 0 0 0 0 0 0 2 2]
1629 [0 0 0 0 0 0 0 2 2]]

```

```

1630

```

```

1631 E_omega column
1632 [j0 + 4*j7 + 4*j8]
1633 [          ]
1634 [    j0    ]
1635 [          ]
1636 [ j1 + j3 + 2*j7 ]
1637 [          ]
1638 [    j1 + j2    ]
1639 [          ]
1640 [ j2 + j4 + 2*j8 ]
1641 [          ]
1642 [    j3 + j4    ]
1643 [          ]
1644 [ j1 + j3 + 2*j7 ]
1645 [          ]
1646 [    j1 + j2    ]
1647 [          ]
1648 [ j2 + j4 + 2*j8 ]
1649 [          ]
1650 [    j3 + j4    ]
1651 [          ]
1652 [ j6 + j7 + j8 ]
1653 [          ]
1654 [    j6        ]
1655 [          ]
1656 [ j5 + j7 + j8 ]
1657 [          ]
1658 [    j5        ]
1659 [          ]
1660 [ j5 + j7 + j8 ]
1661 [          ]
1662 [    j5        ]
1663 [          ]
1664 [ j6 + j7 + j8 ]
1665 [          ]
1666 [    j6        ]
1667 [          ]

```

```

1668 [ 2*j7 + 2*j8 ]
1669 [ ]
1670 [ 2*j7 + 2*j8 ]
1671
1672 V(J) Matrix
1673 [ -j1 - 2*j2 - j4 - 2*j6 - 2*j8          j2 - j3 + 2*j8          -j6 - j
1674 [
1675 [          j2 - j3 + 2*j8          -j1 - 2*j2 - j4 - 2*j5 - 2*j8          -j5 - j
1676 [
1677 [          -j6 - j7 - j8          -j5 - j7 - j8          -j5 - j6 -
1678 [
1679 [          -j6 - j7 - j8          -j5 - j7 - j8          0
1680 [
1681 [j1 + j2 + j3 + j4 - 2*j7 - 2*j8  j1 + j2 + j3 + j4 - 2*j7 - 2*j8          0
1682
1683 7 - j8          -j6 - j7 - j8          j1 + j2 + j3 + j4 + 2*j7 + 2*j8
1684
1685 7 - j8          -j5 - j7 - j8          j1 + j2 + j3 + j4 + 2*j7 + 2*j8
1686
1687 2*j7 - 2*j8          0          0
1688
1689          -j5 - j6 - 2*j7 - 2*j8          0
1690
1691          0          -j0 - 2*j1 - 2*j2 - 2*j3 - 2*j4 - 4*j7 -
1692
1693 ]
1694 ]
1695 ]
1696 ]
1697 ]
1698 ]
1699 ]
1700 ]
1701 4*j8]
1702
1703 V(J) matrix reshaped to show clearly symetry [A B; B A] in Matrix
1704 [ -j1 - 2*j2 - j4 - 2*j6 - 2*j8          -j6 - j7 - j8          j2 - j3 + 2*
1705 [
1706 [          -j6 - j7 - j8          -j5 - j6 - 2*j7 - 2*j8          -j5 - j7 -
1707 [
1708 [          j2 - j3 + 2*j8          -j5 - j7 - j8          -j1 - 2*j2 - j4 - 2
1709 [
1710 [          -j6 - j7 - j8          0          -j5 - j7 -
1711 [
1712 [j1 + j2 + j3 + j4 - 2*j7 - 2*j8          0          j1 + j2 + j3 + j4 -
1713
1714 j8          -j6 - j7 - j8          j1 + j2 + j3 + j4 + 2*j7 + 2*j8
1715
1716 j8          0          0
1717

```

```

1718 *j5 - 2*j8      -j5 - j7 - j8      j1 + j2 + j3 + j4 + 2*j7 + 2*j8
1719
1720 j8      -j5 - j6 - 2*j7 - 2*j8      0
1721
1722 2*j7 - 2*j8      0      -j0 - 2*j1 - 2*j2 - 2*j3 - 2*j4 - 4*j7 -
1723
1724 ]
1725 ]
1726 ]
1727 ]
1728 ]
1729 ]
1730 ]
1731 ]
1732 4*j8]
1733
1734 'A' Matrix
1735 [-j1 - 2*j2 - j4 - 2*j6 - 2*j8      -j6 - j7 - j8      ]
1736 [
1737 [      -j6 - j7 - j8      -j5 - j6 - 2*j7 - 2*j8]
1738
1739 'B' Matrix
1740 [j2 - j3 + 2*j8  -j6 - j7 - j8]
1741 [
1742 [-j5 - j7 - j8      0      ]
1743
1744 New V(J) Matrix (A - B)
1745 [-j1 - 3*j2 + j3 - j4 - 2*j6 - 4*j8      0      ]
1746 [
1747 [      j5 - j6      -j5 - j6 - 2*j7 - 2*j8]
1748
1749
1750 ---- Mineur Analysis ----
1751 Limiting mineur analysis to mineurs of size smaller or equal to 5
1752
1753 Mineur:
1754 Poly(j1 + 3*j2 - j3 + j4 + j5 + 3*j6 + 2*j7 + 6*j8, j1, j2, j3, j4, j5, j6, j7
1755 , j8, domain='ZZ')
1756 Compounds number: [(0,), (1,)]
1757
1758 Mineur:
1759 Poly(j1*j5 + j1*j6 + 2*j1*j7 + 2*j1*j8 + 3*j2*j5 + 3*j2*j6 + 6*j2*j7 + 6*j2*j8
1760 - j3*j5 - j3*j6 - 2*j3*j7 - 2*j3*j8 + j4*j5 + j4*j6 + 2*j4*j7 + 2*j4*j8 + 2*j
1761 5*j6 + 4*j5*j8 + 2*j6**2 + 4*j6*j7 + 8*j6*j8 + 8*j7*j8 + 8*j8**2, j1, j2, j3,
1762 j4, j5, j6, j7, j8, domain='ZZ')
1763 Compounds number: [(0, 1)]
1764
1765 -----
1766 Restrictions to solve
1767 [j0 > 0,

```

```

1768  j1 > 0,
1769  j2 > 0,
1770  j3 > 0,
1771  j4 > 0,
1772  j5 > 0,
1773  j6 > 0,
1774  j7 > 0,
1775  j8 > 0,
1776  -j1 - 3*j2 + j3 - j4 - j5 - 3*j6 - 2*j7 - 6*j8 > 0,
1777  -j1*j5 - j1*j6 - 2*j1*j7 - 2*j1*j8 - 3*j2*j5 - 3*j2*j6 - 6*j2*j7 - 6*j2*j8 + j3*j5 +
1778  j3*j6 + 2*j3*j7 + 2*j3*j8 - j4*j5 - j4*j6 - 2*j4*j7 - 2*j4*j8 - 2*j5*j6 - 4*j5*j8 -
1779  2*j6**2 - 4*j6*j7 - 8*j6*j8 - 8*j7*j8 - 8*j8**2 > 0]
1780
1781  Trying to find a solution with the restrictions
1782
1783  Found a solution on try number 4 of 10000
1784
1785  (Randomly) Chosen variable values:
1786  [j6: 1.62297451374,
1787  j8: 0.945971762709,
1788  j5: 0.910672504215,
1789  j7: 1.9686725392,
1790  ]
1791
1792  Values that variables can take given the already chosen (see above) values for variables
1793  [j8, j6, j5, j7]
1794  [j1 = (0, oo)
1795  j2 = (0, oo)
1796  j3 = (Max(j1 + 3*j2 + 153927717000673/10000000000000, j1 +
1797  2508880686527950*j2/836293562175983 + 5879006655441000/836293562175983), oo)
1798  j4 = (0, Min(-j1 - 2508880686527950*j2/836293562175983 + j3 - 5879006655441000/836293562175983,
1799  -j1 - 3*j2 + j3 - 153927717000673/10000000000000))]
1800
1801  Sampling 10 points from js intervals
1802  Saving sampled states to 'samples/iwamoto-imperfect'
1803  *****
1804  ***  Iwamoto-Imperfect  ***
1805  *****

```

1806 The Iwamoto model under imperfect conditions figure

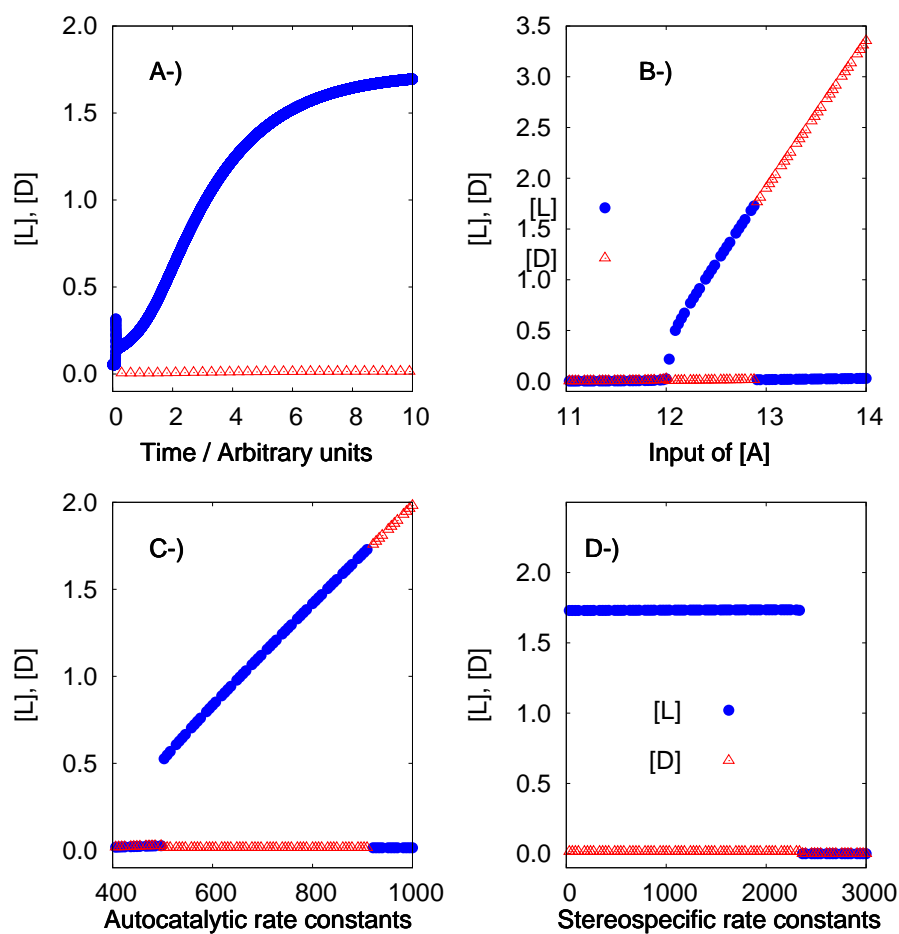


Figure S1. The Iwamoto model under imperfect conditions. A-) Time series; and bifurcation diagrams for B-) Input flow reaction $P \xrightarrow{k_0} A$, C-) autocatalytic reactions $A + L \xrightarrow{k_2} 2L$ and $A + D \xrightarrow{k_2} 2D$, and D-) stereospecific reactions $L + E_L \xrightarrow{k_6} Z_L$ and $D + E_D \xrightarrow{k_8} Z_D$. The initial concentrations of the enantiomers L and D were equals until the 15th decimal position, equivalent to an enantiomeric excess $ee = \frac{[L] - [D]}{[L] + [D]} = 7.051936 \times 10^{-15}$.