If in Noise\_getCommunityParam, we set alpha <- runif(N,min = 0,max = 1),

c0 <- matrix(runif(N\*N, min = -0.08, max = 0.02),nrow = N)

Running a simulation of 6 member community, it shows the species-wise density at the steady state:

Var1 Var2 Var3 Var4 Var5 Var6

44 0.0000000 1.323396 0.0000000 0.0000000 0.0000000 1.0943338

26 0.7438431 0.000000 0.0000000 0.0000000 0.6862505 0.0000000

6 0.7520049 0.000000 0.6282389 0.0000000 0.0000000 0.0000000

34 0.7281294 0.000000 0.0000000 0.0000000 0.0000000 1.1897887

17 0.0000000 0.000000 0.0000000 0.0000000 0.7614600 0.0000000

25 0.0000000 0.000000 0.0000000 0.1957405 0.7578645 0.0000000

32 0.0000000 1.431469 0.0000000 0.0000000 0.5925438 0.0000000

15 0.0000000 1.401099 0.4740393 0.0000000 0.0000000 0.0000000

3 0.0000000 1.460247 0.0000000 0.0000000 0.0000000 0.0000000

58 0.7281228 0.000000 0.0000000 0.0000000 0.0000000 1.1898125

14 0.7520058 0.000000 0.6281989 0.0000000 0.0000000 0.0000000

35 0.0000000 1.323391 0.0000000 0.0000000 0.0000000 1.0943390

24 0.0000000 1.401094 0.4740423 0.0000000 0.0000000 0.0000000

22 0.7520107 0.000000 0.6281987 0.0000000 0.0000000 0.0000000

38 0.7281272 0.000000 0.0000000 0.0000000 0.0000000 1.1897523

62 0.7281236 0.000000 0.0000000 0.0000000 0.0000000 1.1897613

51 0.0000000 1.323391 0.0000000 0.0000000 0.0000000 1.0943406

53 0.0000000 0.000000 0.4383319 0.0000000 0.4448151 0.9210552

16 0.0000000 1.401095 0.4740392 0.0000000 0.0000000 0.0000000

13 0.0000000 0.000000 0.6079449 0.2562741 0.0000000 0.0000000

20 0.0000000 1.431475 0.0000000 0.0000000 0.5925484 0.0000000

29 0.0000000 0.000000 0.6215705 0.0000000 0.7314397 0.0000000

37 NaN 0.000000 NaN 0.0000000 0.0000000 NaN

45 0.0000000 0.000000 0.0000000 0.2286121 0.0000000 1.1880353

7 0.0000000 1.401044 0.4740915 0.0000000 0.0000000 0.0000000

18 0.7438516 0.000000 0.0000000 0.0000000 0.6862976 0.0000000

31 0.0000000 1.431469 0.0000000 0.0000000 0.5925495 0.0000000

43 0.0000000 1.323393 0.0000000 0.0000000 0.0000000 1.0943376

41 0.0000000 0.000000 0.0000000 0.2287522 0.0000000 1.1880666

55 0.0000000 1.323393 0.0000000 0.0000000 0.0000000 1.0943382

23 0.0000000 1.401099 0.4740392 0.0000000 0.0000000 0.0000000

5 0.0000000 0.000000 0.6204166 0.0000000 0.0000000 0.0000000

28 0.0000000 1.431792 0.0000000 0.0000000 0.5925510 0.0000000

19 0.0000000 1.431425 0.0000000 0.0000000 0.5925932 0.0000000

10 0.7587323 0.000000 0.0000000 0.1544143 0.0000000 0.0000000

57 0.0000000 0.000000 0.0000000 NaN 0.1749793 NaN

8 0.0000000 1.401098 0.4740410 0.0000000 0.0000000 0.0000000

2 0.7790359 0.000000 0.0000000 0.0000000 0.0000000 0.0000000

64 0.0000000 1.323394 0.0000000 0.0000000 0.0000000 1.0943367

47 0.0000000 1.323391 0.0000000 0.0000000 0.0000000 1.0943402

4 0.7276053 1.358469 0.0000000 0.0000000 0.0000000 0.0000000

60 0.0000000 1.323399 0.0000000 0.0000000 0.0000000 1.0943403

46 0.7281240 0.000000 0.0000000 0.0000000 0.0000000 1.1897711

33 0.0000000 0.000000 0.0000000 0.0000000 0.0000000 1.2047271

54 0.7281273 0.000000 0.0000000 0.0000000 0.0000000 1.1897503

52 0.0000000 1.323391 0.0000000 0.0000000 0.0000000 1.0943392

9 0.0000000 0.000000 0.0000000 0.2463660 0.0000000 0.0000000

36 0.0000000 1.323371 0.0000000 0.0000000 0.0000000 1.0943393

56 0.0000000 1.323395 0.0000000 0.0000000 0.0000000 1.0943360

21 0.0000000 0.000000 0.6216346 0.0000000 0.7314667 0.0000000

1 0.0000000 0.000000 0.0000000 0.0000000 0.0000000 0.0000000

42 0.7281247 0.000000 0.0000000 0.0000000 0.0000000 1.1897880

59 0.0000000 1.323394 0.0000000 0.0000000 0.0000000 1.0943367

40 0.0000000 1.323393 0.0000000 0.0000000 0.0000000 1.0943376

48 0.0000000 1.323391 0.0000000 0.0000000 0.0000000 1.0943398

50 0.7281233 0.000000 0.0000000 0.0000000 0.0000000 1.1897562

63 0.0000000 1.323395 0.0000000 0.0000000 0.0000000 1.0943365

39 0.0000000 1.323398 0.0000000 0.0000000 0.0000000 1.0943335

27 0.0000000 1.431468 0.0000000 0.0000000 0.5925508 0.0000000

11 0.0000000 1.418392 0.0000000 0.2838864 0.0000000 0.0000000

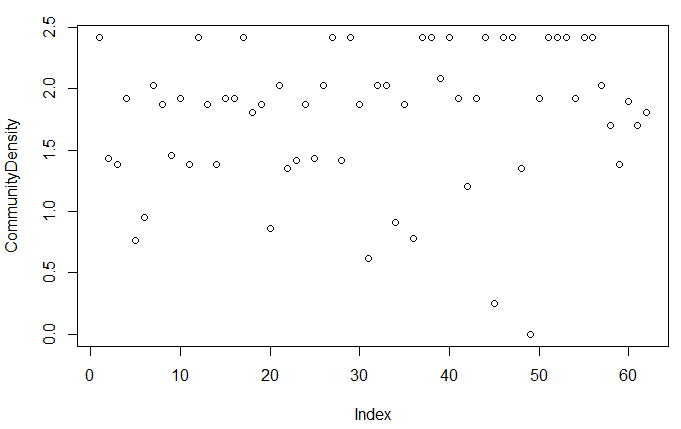
30 0.7520070 0.000000 0.6281923 0.0000000 0.0000000 0.0000000

49 0.7281891 0.000000 0.0000000 0.0000000 0.0000000 1.1668014

12 0.0000000 1.418552 0.0000000 0.2838079 0.0000000 0.0000000

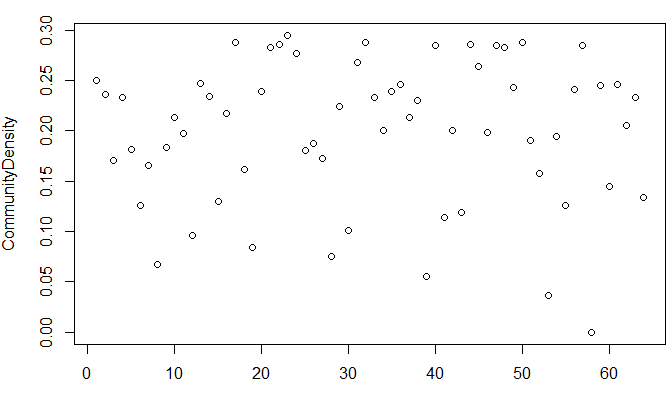
61 0.0000000 0.000000 0.4382265 0.0000000 0.4449179 0.9211375

It seems that as long as a species reached steady state eventually (instead of dying out and get 0 density), the value is stable. E.g. species1 is always about 0.72, species 2 is always about 1.4, etc. This explains the lack of noise in final biomass.



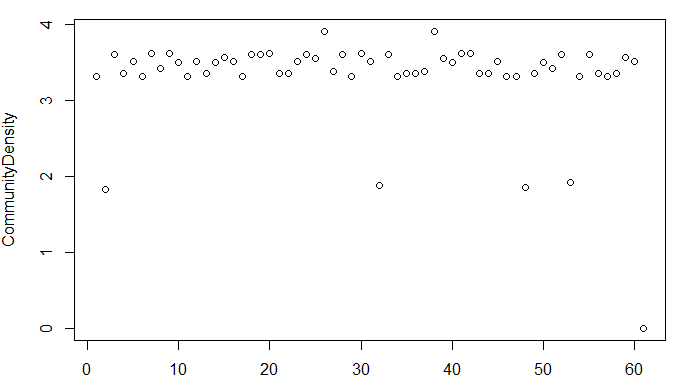
In contrast, if we set

alpha <- runif(N,min = 0,max = 0.1), i.e. let species grow a lot slower, a lot more randomness is present:



However if we narrow the range of species growth rate:

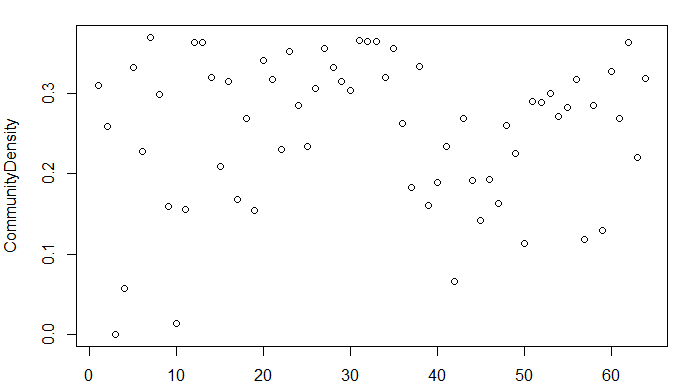
alpha <- runif(N,min = 0.9,max = 1), less randomness is observed



The magnitude of Co on the randomness

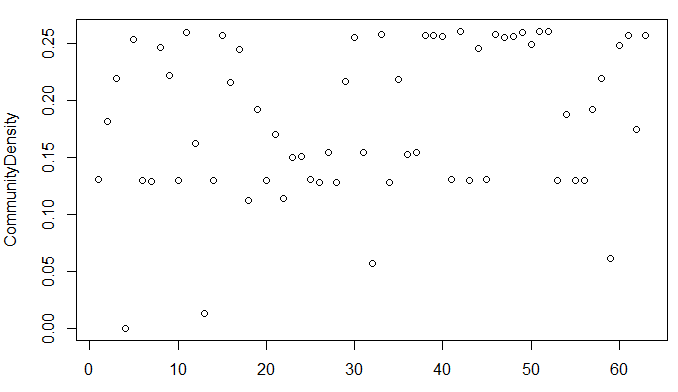
alpha <- runif(N,min = 0,max = 0.1)

c0 <- matrix(runif(N\*N, min = -0.08, max = 0.02),nrow = N)



And if we set

c0 <- matrix(runif(N\*N, min = -0.8, max = 0.2),nrow = N)



Randomness decreased.

HOI has effect on Co (pairwise interaction score), so strength of HOI also affect randomness. This can be qualitatively evaluated by tuning HOI strength and look at the final plot.

When the interaction is strong, those species started out at disadvantage quickly dies out, the remaining combination is limited.