feasible db planning

sqlite vs duckdb —

sqlite is more established. duckdb is the Next Big Thing (perhaps) & looks kind of in-progress.

p_tables
(start w/existing)

fs_draws
(start by saving actual draws? TBD size of dbs. (scrods probably had lots of duplicates))

s	n	rank	abund	draw
...	...	...	...	...

fs_summary_stats

s	n	draw	hill1	hill2
...	...	...	...	...

questions

- comprehensive or organic?
what I mean by this is,
1) trying to sample comprehensively would quickly become ridiculous, also I need for exact match on S & N.
is the thinking that eventually you will hit duplicate values? (yes I think so)
it also seems worth — potentially — exploring a kind of fuzzy-matching of S & N
or if there is some predictability of the distribution of hill #s for ~~the~~ known S & N.

that seems like a potentially solvable phenomenologically problem?

hill1	hill2	hill0	s	n
...	...	...		...

→ ???

(in principle this is a v. simple use case... other generative models are harder. I wonder if you could train a similar system to generate vole predictions)

given s, n

generate, expected hill1, hill2, hill0 DISTRIBUTIONS OF

This would be a very testable & benchmarkable use case.

How much time it would save would depend on how quickly it runs (& how accurately, & how long it takes me to code up)

machine learning + combinatorics → accelerated work in theor. ecology