

feasible db planning

p-tables
(start w/existing)

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

fs_draws

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

s	n	rank	abund	draw
...

questions

fs-summary-stats

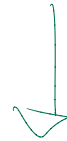
s	n	draw	hill1	hill2
...

- 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 $S \oplus N$.
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 $S \oplus N$
or there is some predictability of the
distribution of hillabs for ~~the~~ known s and n .

that seems like a potentially solvable phenomenologically problem?

hill1	hill2	hill0	s	n
...

→ ???



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

(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
role
predictions)