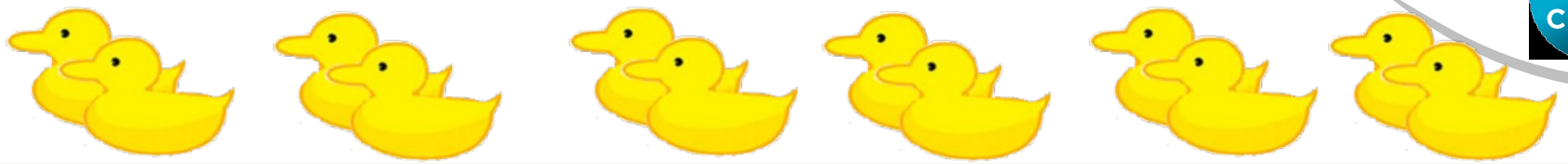
Design for CKMR

Mark Bravington, CSIRO: June 2021

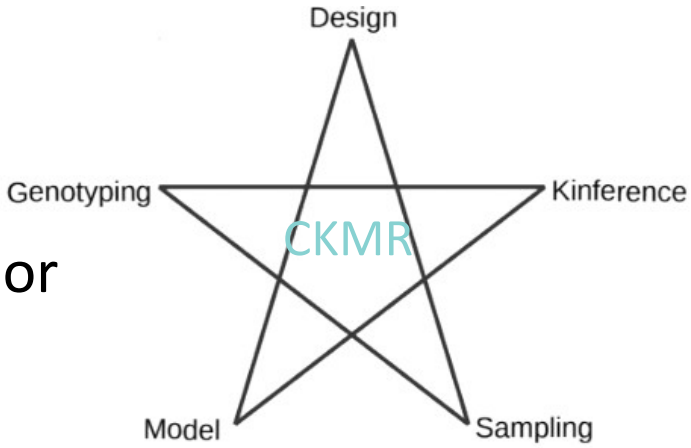


Design in CKMR

If You...

- do not collect *enough* samples, or
- do not collect the right *types* of sample, or
- do not measure their stuff adequately

... **then** CKMR will **fail**

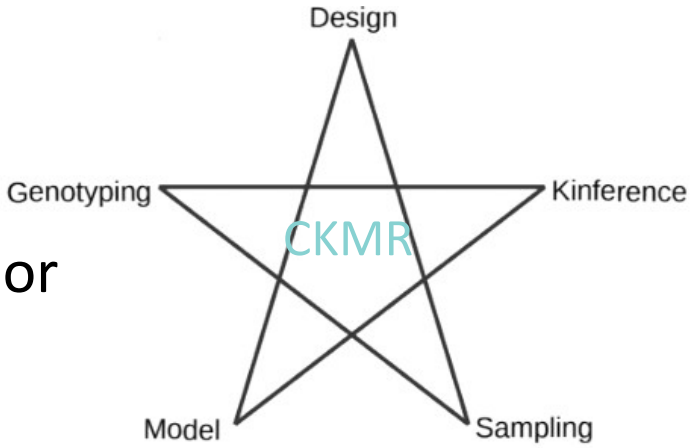


Design in CKMR

If You...

- do not collect *enough* samples, or
- do not collect the right *types* of sample, or
- do not measure their stuff adequately

... **then** CKMR will **fail**



"*enough samples*" really means "enough kin-pairs" ...

Can't control latter, but can plan former and types, and stuff

via... **DESIGN**

Levels of design

0. Qualitative considerations..?
1. Aim for at least XXX kin-pairs
    - by calculating E[total POPs] & E[total HSPs]

    (the laugh test)
2. Variance straight from model code
3. Inverse design
4. Optimal design
5. “Long CKMR”
6. Role of simulation

Design: qualitative sampling needs

What I reckon... for *Fish*

1. POPs **&** HSPs

   ie ads & "juves"

   how old can a "juve" be?

2. Accurate-enuf juve age
3. Ads: age or length both is best

   and sex, of course

4. Full size range of ads
5. Reasonable # juve cohorts
6. Adequate spatial coverage

   to allow for checks; a priori important

What I reckon... for *Mammals*

1. Are you sure?
2. Fish version is best
3. At a pinch, "juve" HSPs only

   or even POPs only-- less likely ?

4. Accurate-enuf juve age
5. Plenty of juve cohorts

   NB eg skip-spawning

6. Adequate spatial coverage

Design: beyond the qualitative

IJAD

KISS

FFS

Design: beyond the qualitative

IJAD

It’s Just A Design...

KISS

FFS

Design: beyond the qualitative

IJAD — It's Just A Design...

KISS — Keep It Simple, Stupid!

FFS

Design: beyond the qualitative

IJAD	It's Just A Design...
KISS	Keep It Simple, Stupid!
FFS	Fast and Fairly Simple

CKMR design: the rock-bottom line

- For a precise estimate*, must find a fair number of kin-pairs
- $\mathbb{P}\left[K_{ij} \neq \mathrm{UP}\right] \sim \bar{N}_{\mathrm{adult}}^{-1}$
- $\hat{\bar{N}}_{\mathrm{adult}} \propto (\#\mathrm{POPs\ and/or\ HSPs})^{-1}$

CKMR design: the rock-bottom line

- For a precise estimate*, must find a fair number of kin-pairs
- $\mathbb{P}\left[K_{ij} \neq \mathrm{UP}\right] \sim \bar{N}_{\mathrm{adult}}^{-1}$
- $\hat{\bar{N}}_{\mathrm{adult}} \propto (\#\mathrm{POPs\ and/or\ HSPs})^{-1}$
- #(POPs and/or HSPs) ~ Poisson; so need *at least* 50 for good CV

CKMR design: the rock-bottom line

- For a precise estimate*, must find a fair number of kin-pairs
- $\mathbb{P}\left[K_{ij} \neq \mathrm{UP}\right] \sim \bar{N}_{\mathrm{adult}}^{-1}$
- $\hat{\bar{N}}_{\mathrm{adult}} \propto (\#\mathrm{POPs\ and/or\ HSPs})^{-1}$
- #(POPs and/or HSPs) ~ Poisson; so need *at least* 50 for good CV
- Sample sizes: m_z with covariate z
- Group "alike" comparisons (same z, z')

CKMR design: the rock-bottom line

- For a precise estimate*, must find a fair number of kin-pairs
- $\mathbb{P}\left[K_{ij} \neq \mathrm{UP}\right] \sim \bar{N}_{\mathrm{adult}}^{-1}$
- $\hat{\bar{N}}_{\mathrm{adult}} \propto (\#\mathrm{POPs\ and/or\ HSPs})^{-1}$
- #(POPs and/or HSPs) ~ Poisson; so need *at least* 50 for good CV

CKMR design: the rock-bottom line

- For a precise estimate*, must find a fair number of kin-pairs
- $\mathbb{P}\left[K_{ij} \neq \mathrm{UP}\right] \sim \bar{N}_{\mathrm{adult}}^{-1}$
- $\hat{\bar{N}}_{\mathrm{adult}} \propto (\#\mathrm{POPs\ and/or\ HSPs})^{-1}$
- #(POPs and/or HSPs) ~ Poisson; so need *at least* 50 for good CV
- Sample sizes: m_z with covariate z

Wanna get 50–100

- Group "alike" comparisons (same z, z')

Each comp has prob ~O(1/*N*)

m samples, so ~m^2 comps

... so m^2 * “const” / *N* = 50–100

... so *m* = “const (a different one)” * sqrt(*N*)

CKMR design: the rock-bottom line

- For a precise estimate*, must find a fair number of kin-pairs
- $\mathbb{P}\left[K_{ij} \neq \mathrm{UP}\right] \sim \bar{N}_{\mathrm{adult}}^{-1}$
- $\hat{\bar{N}}_{\mathrm{adult}} \propto (\#\mathrm{POPs\ and/or\ HSPs})^{-1}$
- #(POPs and/or HSPs) ~ Poisson; so need *at least* 50 for good CV
- Sample sizes: m_z with covariate z
- Group "alike" comparisons (same z, z')

$$\mathbb{E}\left[\#\mathrm{POPs}|\theta_0\right] = \sum_{z,z'} c_{zz'} \times (m_z m_{z'}) \times p_{\mathrm{POP}zz'}(\theta_0)$$

usually 1

guess at demographic params

comps of this "type"

this kinship prob

CKMR design: the rock-bottom line

"*If we sample X of these and Y of those--- will we get enuf kin?*"

Software `microscoping::ckmr_laugh_test` for stock-assessed

It's rough--- but it's ready.

CKMR design: the rock-bottom line

"*If we sample X of these and Y of those--- will we get enuf kin?*"

Software `microscoping::ckmr_laugh_test` for stock-assessed

It's rough--- but it's ready.

"PASSING THE LAUGH TEST" IS NOT "DOING A DESIGN"

CKMR design: total Exp[Kin] is not enough

Even if you get lots of kin-pairs, does not guarantee good CV

- depends on covars
- and measuring stuff adequately!

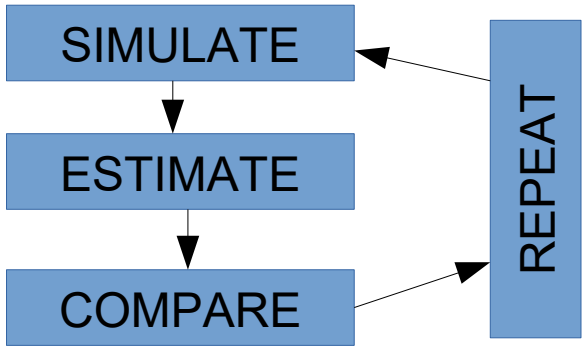
CKMR design: total Exp[Kin] is not enough

Even if you get lots of kin-pairs, does not guarantee good CV

- depends on covars
- and measuring stuff adequately!

- To *properly* evaluate a design: choose a *goal* $g(\theta)$ e.g. trend in biomass
  - check that sample sizes $\{m_{z \in \mathcal{Z}}\}$ will give low enough $\mathbb{V}\left[g(\hat{\theta})\right]$

"Generic" process for evaluation:



That's a **lot** of work !!

CKMR design: an easier way

- Big populations, sparse sampling: pairwise comps independent
- Each pair comp is Poisson: expected Fisher Info is

$$H\left(z,z'\right)=4\left[\frac{d\sqrt{p_{zz'}}}{d\theta}\right]\left[\frac{d\sqrt{p_{zz'}}}{d\theta}\right]^{\top}$$

$$H=H_0+\sum_{z,z'}m_z m_{z'}H\left(z,z'\right)$$

$$\mathbb{V}\left[g\left(\hat{\theta}\right)\right]\approx\left[\frac{dg}{d\theta}\right]^{\top}H^{-1}\left[\frac{dg}{d\theta}\right]$$

Quadratic dependence on sample size

Numerical derivatives are OK for *p* and *g*

NO simulation; NO estimation; NO repetition !

Other data: priors into H_0 ; pure multinomials (age samples)

CKMR design: an easier way

What on does all *that* mean?!

- Big populations, sparse sampling: pairwise comps independent
- Each pair comp is Poisson: expected Fisher Info is

Well, you just need:

$$H(z,z') = 4\left[\frac{d\sqrt{p_{zz'}}}{d\theta}\right]\left[\frac{d\sqrt{p_{zz'}}}{d\theta}\right]^{\top}$$

$$H = H_0 + \sum_{z,z'} m_z m_{z'} H(z,z')$$

Quadratic dependence on sample size

$$\mathbb{V}\left[g\left(\hat{\theta}\right)\right] \approx \left[\frac{dg}{d\theta}\right]^{\top} H^{-1}\left[\frac{dg}{d\theta}\right]$$

Numerical derivatives are OK for *p* and *g*

NO simulation; NO estimation; NO repetition !

Other data: priors into H_0 ; pure multinomials (age samples)

CKMR design: an easier way

What on does all *that* mean?!

- Big populations, sparse sampling: pairwise comps independent
- Each pair comp is Poisson: expected Fisher Info is

$$H(z,z') = 4\left[\frac{d\sqrt{p_{zz'}}}{d\theta}\right]\left[\frac{d\sqrt{p_{zz'}}}{d\theta}\right]^{\top}$$

Well, you just need:

- to code the lglk you expect to use for real but you **can** take shortcuts !

$$H = H_0 + \sum_{z,z'} m_z m_{z'} H(z,z')$$

Quadratic dependence on sample size

$$\mathbb{V}\left[g\left(\hat{\theta}\right)\right] \approx \left[\frac{dg}{d\theta}\right]^{\top} H^{-1} \left[\frac{dg}{d\theta}\right]$$

Numerical derivatives are OK for *p* and *g*

NO simulation; NO estimation; NO repetition !

Other data: priors into H_0 ; pure multinomials (age samples)

CKMR design: an easier way

What on does all *that* mean?!

- Big populations, sparse sampling: pairwise comps independent
- Each pair comp is Poisson: expected Fisher Info is

$$H(z,z') = 4\left[\frac{d\sqrt{p_{zz'}}}{d\theta}\right]\left[\frac{d\sqrt{p_{zz'}}}{d\theta}\right]^{\top}$$

$$H = H_0 + \sum_{z,z'} m_z m_{z'} H(z,z')$$

$$\mathbb{V}\left[g\left(\hat{\theta}\right)\right] \approx \left[\frac{dg}{d\theta}\right]^{\top} H^{-1} \left[\frac{dg}{d\theta}\right]$$

Well, you just need:

- to code the lglk you expect to use for real but you **can** take shortcuts !
- prior guesstimates of pop dyn stuff eg from stock assessment

Quadratic approximation on sample size

Numerical derivatives are OK for *p* and *g*

NO simulation; NO estimation; NO repetition !

Other data: priors into H_0 ; pure multinomials (age samples)

CKMR design: an easier way

What on does all *that* mean?!

Well, you just need:

- to code the lglk you expect to use for real
- prior guesstimates of pop dyn stuff
- some idea of what you *want* to find out!

but you **can** take shortcuts !

eg from stock assessment

Numerical derivatives are OK for *p* and *g*

NO simulation; NO estimation; NO repetition !

Other data: priors into H_0 ; pure multinomials (age samples)

CKMR design: an easier way

What on does all *that* mean?!

Well, you just need:

- to code the lglk you expect to use for real but you **can** take shortcuts !
- prior guesstimates of pop dyn stuff eg from stock assessment
- some idea of what you *want* to find out!
- sample sizes / sampling schemes to try out

Numerical derivatives are OK for *p* and *g*

NO simulation; NO estimation; NO repetition !

Other data: priors into H_0 ; pure multinomials (age samples)

CKMR design: an easier way

What on does all *that* mean?!

Well, you just need:

- to code the lglk you expect to use for real
- prior guesstimates of pop dyn stuff
- some idea of what you *want* to find out!
- sample sizes / sampling schemes to try out

but you **can** take shortcuts !

eg from stock assessment

You **don't** need to simulate* nor to actually *fit* your model;
there's a generic "glue framework" that does the maths

CKMR design: NAtl makos

CKMR design: NAtl makos etc

Q: *But that's just under one scenario about "truth". Shouldn't I explore others?*

A: Arguably, that's not really necessary...

CKMR design: Inverse design

What if *no* prior stock assessment ?

eg SAtl makos; Blue-Eye Trevalla

Optimal design

- Everybody knows: "*don't trust optimal designs*"
- Not always appropriate to try

NEVERTHELESS

- it lets you know when to stop trying
- "CKMR design space" is a big place. Optimal designs are *beacons*
  - ie several optimal designs for different goals
- You learn a lot about where the "information" comes from

Optimal design for CKMR

$$H(z,z') = 4\left[\frac{d\sqrt{p_{zz'}}}{d\theta}\right]\left[\frac{d\sqrt{p_{zz'}}}{d\theta}\right]^{\top}$$

$$H = H_0 + \sum_{z,z'} m_z m_{z'} H(z,z')$$

$$\mathbb{V}\left[g\left(\hat{\theta}\right)\right] \approx \left[\frac{dg}{d\theta}\right]^{\top} H^{-1}\left[\frac{dg}{d\theta}\right]$$

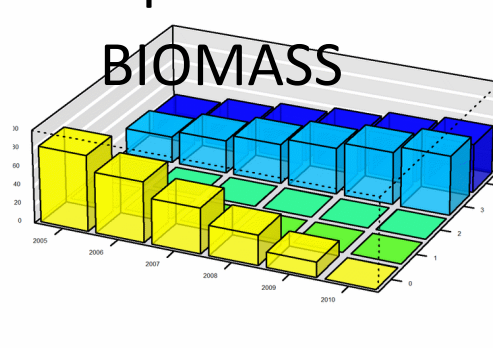
$$V(m) \triangleq \mathbb{V}\left[\hat{g}(\theta)\,|m\right]$$

- *H* is sum of outer products: closed form inverse Sherman-Woodbury
- d*V*/d*m* , d²*V*/d*m*² easily found by Automatic Differentiation
- Cost constraint (or target variance for minimum cost)
- Replace *V*(*m*) by 2nd order approx; solve Quadratic Program; repeat
  - reliable and quick
  - NB *non-convex* QP

Optimal design: example

- "Fish-like": age structure, fecundity depends on age, survival depends on year
- Cross-cohort HSPs with 0-year-olds; POPs (all age classes)
- 10-year trend in abundance, 5 ages (5 fixed effects, 10 random effects)

Opt. for BIOMASS

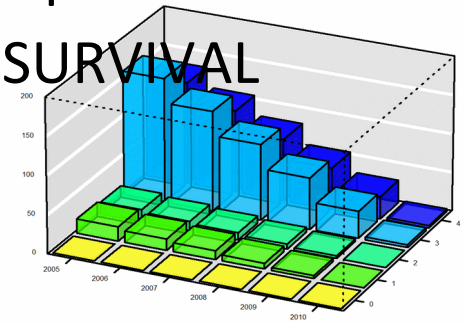


At these two optima :

3X variance (biomass)

2X variance (survival)

Opt. for SURVIVAL



- Optimal design does depend on goal
- Runtime: ~5 seconds

Summary

- CKMR can handle big problems (big population sizes)
  - but you'll need a lot of samples
- *Bare minimum* due diligence: will your design give enough kin-pairs ?
  - need to compute kinship probs based on a guess at parameters
  - maybe `ckmr_laugh_test` will do...
  - or maybe you need to code up your planned lglk. If so, then...
- ... not much extra work to properly evaluate a design
  - just supply a goal function (in R is OK)
- & almost no extra work to consider *optimal* designs

Design in CKMR: niggling concerns laid to rest

Q: I have 1000 samples but 1000*1000/2 comparisons. Are they *really* all independent? Really?

A: Yeah pretty much, prolly.

At least, if yr popn is Big ”sparse sampling”

Design in CKMR: niggling concerns laid to rest

Q: I have 1000 samples but 1000*1000/2 comparisons. Are they *really* all independent? Really?

A: Yeah pretty much, prolly.

At least, if yr popn is Big ”sparse sampling”

How to approach the thought-experiment:

- think about Maternal descent Qs ie separately from P descent
- implications & chance of knowing result of “earlier” comp

Design in CKMR: niggling concerns laid to rest

Q: I have 1000 samples but 1000*1000/2 comparisons. Are they *really* all independent? Really?

A: Yeah pretty much, prolly.

At least, if yr popn is Big ”sparse sampling”

How to approach the thought-experiment:

- think about Maternal descent Qs ie separately from P descent
- implications & chance of knowing result of “earlier” comp

Post hoc diagnostic: # triads etc.

Larval sampling? different, tricky; *not* the first resort!

“Long CKMR”

To date, fisheries interest in CKMR has come from The Desperate

- “how on earth can we ground-truth the assessment?”

Then... lots of work to get to that point.

But what happens next?!

“Long CKMR”

To date, fisheries interest in CKMR has come from The Desperate

- “how on earth can we ground-truth the assessment?”

Then... lots of work to get to that point.
But what happens next?!

CKMR is easy once you’ve gotten your first “N”
?Use in ongoing MPs?
Needs simulation--- can’t do via nifty variance calcs

Role of simulation in CKMR

MUCH LESS THAN YOU THINK!

- You *don't* need simulation for most Designs
  - and you *shouldn't* do it because it will waste time

Role of simulation in CKMR

MUCH LESS THAN YOU THINK!

- You *don't* need simulation for most Designs
    - and you *shouldn't* do it because it will waste time

- IBM sim *is* required if you plan to fit a very dicey model...

- CKMR IBMs are somewhat painful
    - tho packages can help: `fishSim`, `kinsimmer`, `CKMRsim`

Role of simulation in CKMR

MUCH LESS THAN YOU THINK!

- You *don't* need simulation for most Designs
    - and you *shouldn't* do it because it will waste time

- IBM sim *is* required if you plan to fit a very dicey model...

- CKMR IBMs are somewhat painful
    - tho packages can help: `fishSim`, `kinsimmer`, `CKMRsim`

- For MSE, can just simulate "from" the CKMR model:

    `dpois() ---> qpois()`

- OK: everyone is allowed to do *one* sim... if you must...