First experiments with a new climate model

Jonathan Rougier

Accomplices: David Cameron, Neil Edwards, Andrew Price

25 September, 2006

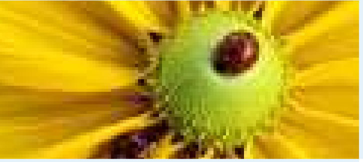
The GENIE-I climate model

`GENIE-I` = `GOLDSTEIN ocean` + `EBM atmosphere` + `sea-ice`

- Our version has an untried resolution of 64×32; this was chosen to help us match up to the atmospheric model of GENIE-II.
- Forcing:
  1. Spin up at pre-industrial $\mathtt{CO}_2$;
  2. Historic forcing up to the present;
  3. 1%pa compound out to 2100.
- Model outputs:
  - Current maximum Atlantic stream-function;
  - Global atmospheric temperatures, 1900, 1950, 2000;
  - THC in 2100 (max, min, three locations);
  - Atmospheric temperature in 2100 (five locations).

Uncertain model inputs

Input	Unit	ID	Min	Max	Mapping $a, \pm, \lambda$
Windstress scaling factor		WSF	1.0	3.0	$0, +, 1$
Ocean horizontal diffusivity	m2s-1	OHD	300	10000	$0, +, 0.5$
Ocean vertical diffusivity	m2s-1	OVD	2.0e-6	2.0e-4	$0, +, 0.5$
Ocean inverse drag coefficient	days	ODC	0.5	5.0	$0, +, 0.5$
Atmospheric heat diffusivity	m2s-1	AHD	1.0e6	1.0e7	$0, +, 0.5$
Atmospheric moisture diffusivity	m2s-1	AMD	5.0e4	5.0e6	$0, +, 0.5$
"Width" of atmospheric heat diffusivity	radians	WAH	0.5	2.0	$0, +, 1$
Zonal heat advection factor		ZHA	0.0	1.0	$0, +, 1$
Zonal / meridional moisture advection		ZMA	0.0	1.0	$0, +, 1$
Sea ice diffusivity	m2s-1	SID	0.3e3	25e3	$0, +, 0.5$
Scaling for Atlantic-Pacific moisture flux	$\times 0.32$ Sv	APM	0.0	2.0	$0, +, 1$
Threshold humidity, for precipitation	%	THP	0.8	0.9	$0, +, 1$
"Climate sensitivity", CO2 radiative forcing	Wm-2	CRF	4.77	6.77	$0, +, 1$
Solar constant	Wm-2	SOC	1363	1373	$0, +, 1$
Greenland melt rate due to global warming	Sv degC-1	GMR	0.01	0.03	$0, +, 1$
Velocity relaxation		REL	0.75	0.95	$1, -, 0.5$



Mapping the inputs

- We will want to represent the model $x \mapsto g(x)$ as

$$g(x) = \sum\nolimits_i \beta_i \, h_i(x) + \text{small residual}$$

where the $h_i(\cdot)$ are specified non-linear functions of x.

- One simple approach is to transform the model inputs univariately, and then use monomials in the transformed inputs for the $h_i(\cdot)$, i.e.,

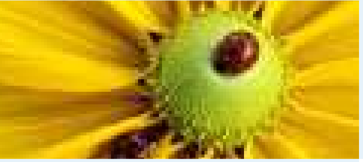
$$x_j \longrightarrow u_j \quad \text{and} \quad h_i(x) = \prod\nolimits_j (u_j)^{r_{ij}}$$

for specified $\{r_{ij}\}$.

- Our mapping is

$$x_j \longrightarrow_{\text{Box-Cox}} \frac{(a \pm x_j)^\lambda - 1}{\lambda} \longrightarrow_{\text{linear}} u_j \in [-1, 1]$$

for specified $(a, \pm, \lambda)_j$. Our choices (not set in stone) are shown in the Inputs Table.



Mapping the inputs (cont)

- We generate designs that are equally-spaced in u_j for each input, and then map these back into x_j in order to evaluate the model.

  Hot tip! It's very important to save the designs in the original units, in case we decide later on to modify the mapping. We also saved the `md5sum` of each evaluation, to help match up the inputs and outputs.

- If our designs are reasonably orthogonal in the u_j, our regressors $h_i(x)$ will be reasonably orthogonal too. Choosing $u_j \in [-1, 1]$ ensures that *even functions* are approximately orthogonal to *odd functions*.

- Orthogonality in the set of all possible regressors is an important feature if we want to explore a number of different collections of regressors, for building parsimonious statistical representations of $g(\cdot)$.

Exploratory designs

Also known as *screening.*

- *Latin hypercubes (LHCs)* provide a good compromise between coverage and detail (for main effects and low-order interactions). They are very cheap to generate.
- For any particular $n \times p$, the *maximin LHC* is a good (deterministic) choice; typically we tend just to simulate a large number of random LHCs and pick the one with the largest minimum interpoint distance.
- We prioritise what we believe are the important inputs by assigning them to the D-optimal subset of columns in our best LHC.

Exploratory designs

Also known as *screening.*

- *Latin hypercubes (LHCs)* provide a good compromise between coverage and detail (for main effects and low-order interactions). They are very cheap to generate.
- For any particular $n \times p$, the *maximin LHC* is a good (deterministic) choice; typically we tend just to simulate a large number of random LHCs and pick the one with the largest minimum interpoint distance.
- We prioritise what we believe are the important inputs by assigning them to the D-optimal subset of columns in our best LHC.

!! But when we evaluated the model over our initial designs we got a very high number of failures :-(

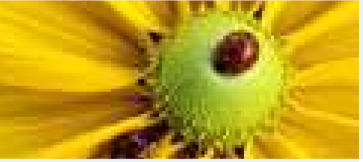
So why does the model fail?

This is pretty much how the experiment progressed . . .

- We knew we were going to put the solver under a lot of stress with extreme input values: our input space has *lots* ($\sim 66,000$) of corners! But failure rates of c50% were a bit much.

- We reckoned the probable cause of failure (at this new higher resolution) was large gradients around the poles, but nothing much could be inferred from simple plots of successes / failures by paired inputs. [failures] [successes]

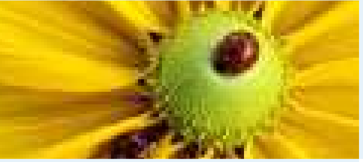
- Neil was off on holiday. David and I suspected the low diffusivities, and so raised their minimum values and modified the curvature of the mapping for the next experiment.



So why does the model fail?

This is pretty much how the experiment progressed . . .

- We knew we were going to put the solver under a lot of stress with extreme input values: our input space has *lots* ($\sim 66,000$) of corners! But failure rates of c50% were a bit much.
- We reckoned the probable cause of failure (at this new higher resolution) was large gradients around the poles, but nothing much could be inferred from simple plots of successes / failures by paired inputs. [failures] [successes]
- Neil was off on holiday. David and I suspected the low diffusivities, and so raised their minimum values and modified the curvature of the mapping for the next experiment.
- This was not a huge success: the failure rate stayed high, and Neil was a bit miffed. So we bit the bullet and decided to do a detailed statistical analysis of what was causing the failures.



Causes of model failure

- *We proceed on the basis that the Southampton cluster was performing correctly.*

- We have $2,500$ evaluations from three LHCs, of which $1,478$ were successful. These evaluations span most of the input space, with lower density around some of the edges.

- We perform a logistic regression analysis relating the outcome, $\{\text{failure}, \text{success}\}$, to regressors of the inputs, i.e.,

$$\text{logit}\,\Pr(\text{success} \mid x) = \sum\nolimits_i \beta_i\, h_i(x) + \text{small residual.}$$

- We use model-building techniques to explore the space of possible regressors in an efficient way.

- We summarise the results graphically to identify clusters of inputs that interact.

Model-building

1. Collections of monomials: An and Owen (2001) provide a useful way of specifying collections of monomials $\prod_j (u_j)^{r_j}$ in terms of (d, w, m), where

$$\sum_j r_j \leq d \qquad \sum_j 1_{r_j > 0} \leq w \qquad \max_j r_j \leq m$$

   We used $(d = 3, w = 3, m = 3)$, giving 1771 candidate regressors (ouch!).

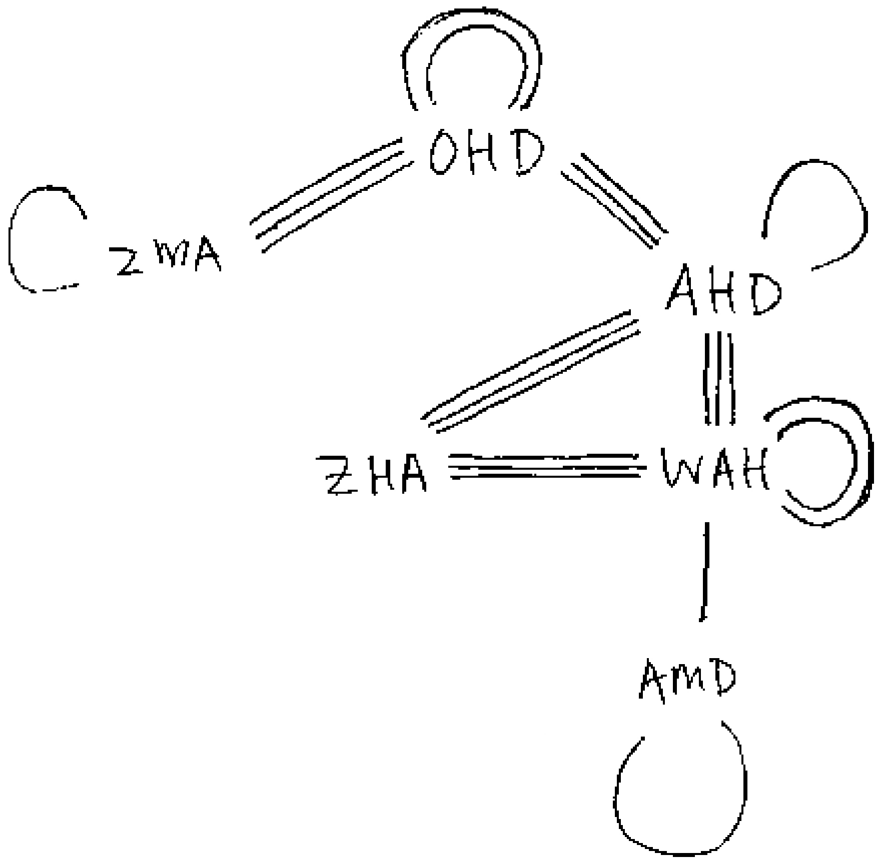
2. Penalising high-order terms. Wary of over-fitting, we insist that high-order terms had to be good enough to justify their lower-order parents, e.g., no $(u_j)^3$ without a $(u_j)^2$, similarly for interaction terms.

3. A cunning wrinkle: we include orthogonal dummy regressors to check for overshooting (due to Mike McKay at LANL).

4. My favoured approach: linear backwards (AIC), everything (AIC), everything (BIC).

Graphical representation of the clusters

Each input is a vertex, and two vertices share an edge for each time they occur together in a monomial.



Visualisation of the probability field

- The statistical model allows us to compute a probability field

$$\mathsf{Pr}(\mathsf{success} \mid x) \qquad \text{for any } x.$$

This is $(16+1)$-dimensional: not very easy to inspect, unless you are from the hyper-dimensional planet Zorg.

- We want low-dimensional projections so that we can understand the causes of model failure, and relate them back to the physics.

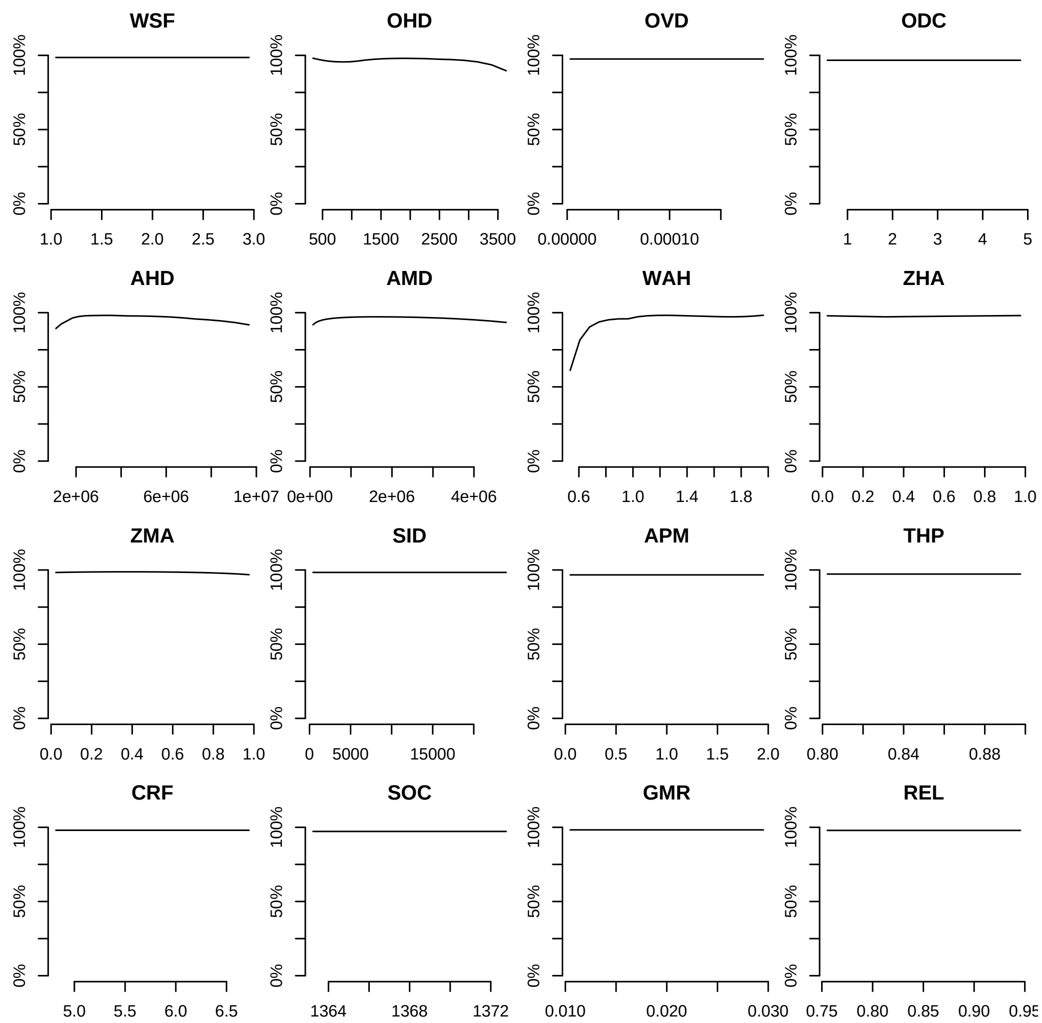
- The conservative projection is

$$\mathsf{Pr}(\mathsf{success} \mid x_I) = \max_{x' \in x \setminus x_I} \mathsf{Pr}\big(\mathsf{success} \mid (x_I, x')\big)$$

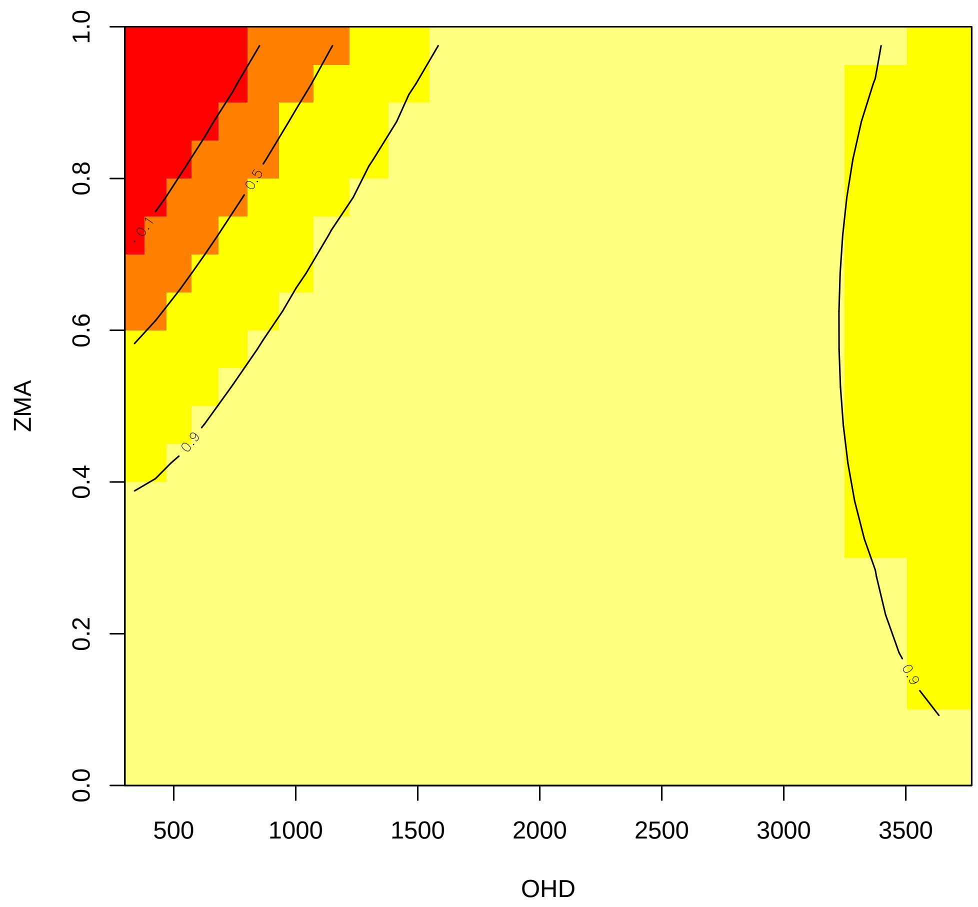
which allows us to identify low values in x_I which are strongly associated with failure.

- If we look at the projections of our main input clusters we should see something interesting . . .

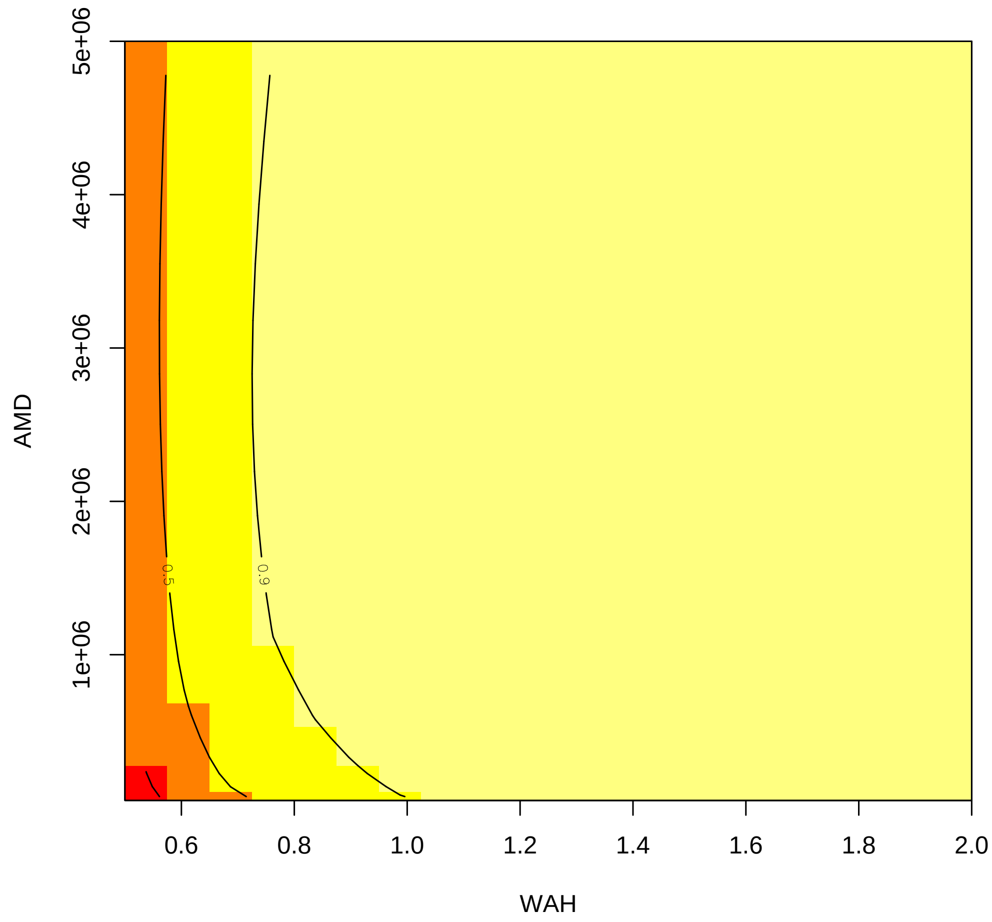
1D margins, all inputs

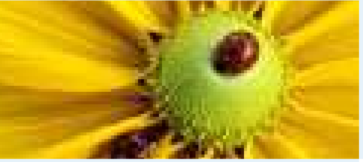


2D: OHD and ZMA

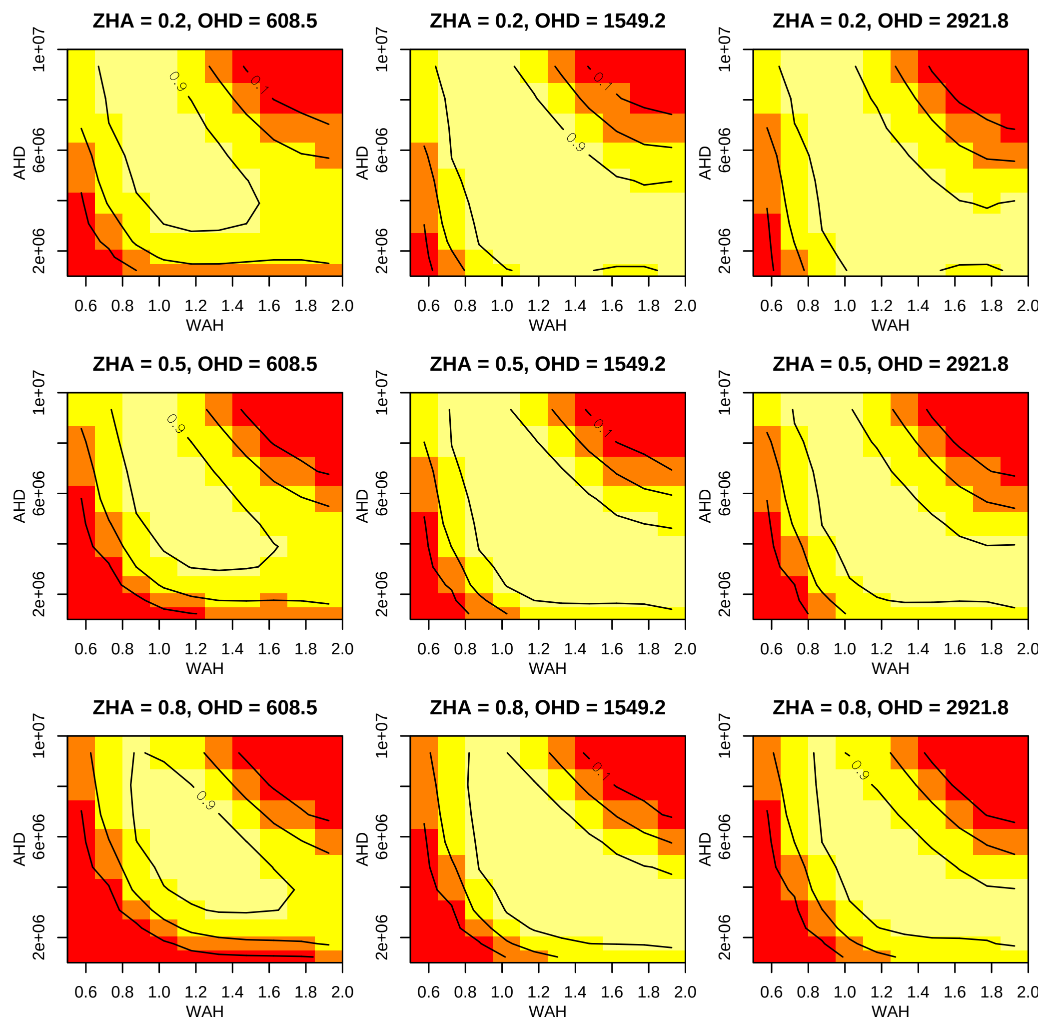


2D: WAH and AMD

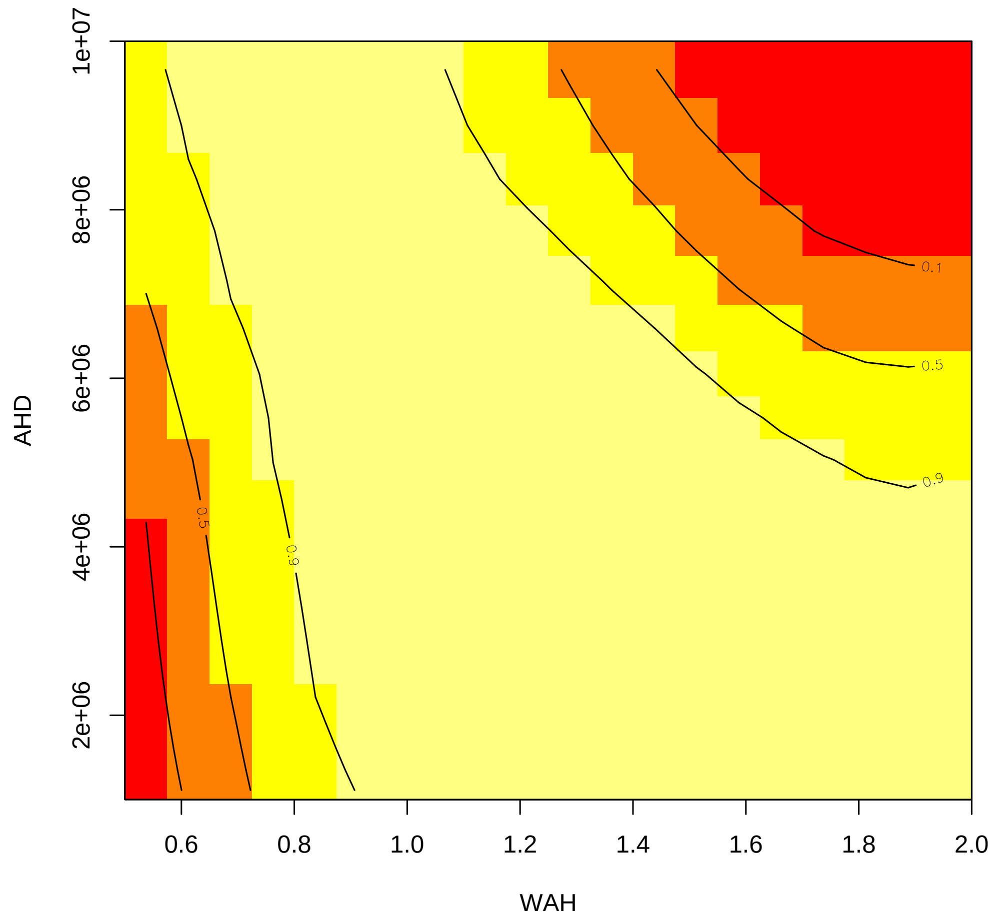




4D: WAH, AHD, ZHA, and OHD



2D: WAH and AHD



Filtering future evaluations

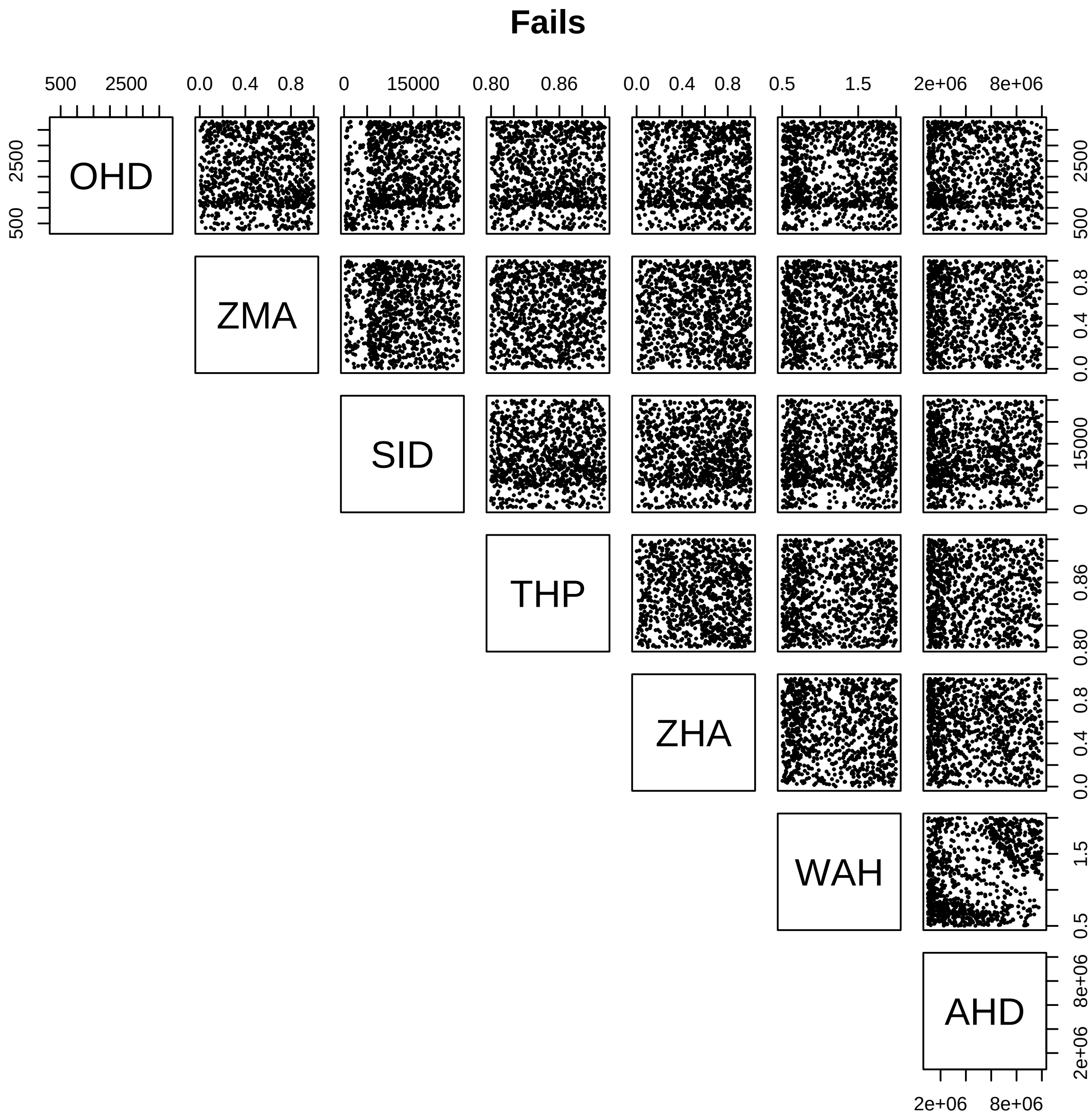
- We can try out different probability thresholds $\Pr(\text{success} \mid x) \geq \nu$ for determining whether any particular choice of x should actually be evaluated.

- We have to balance the two different types of error:

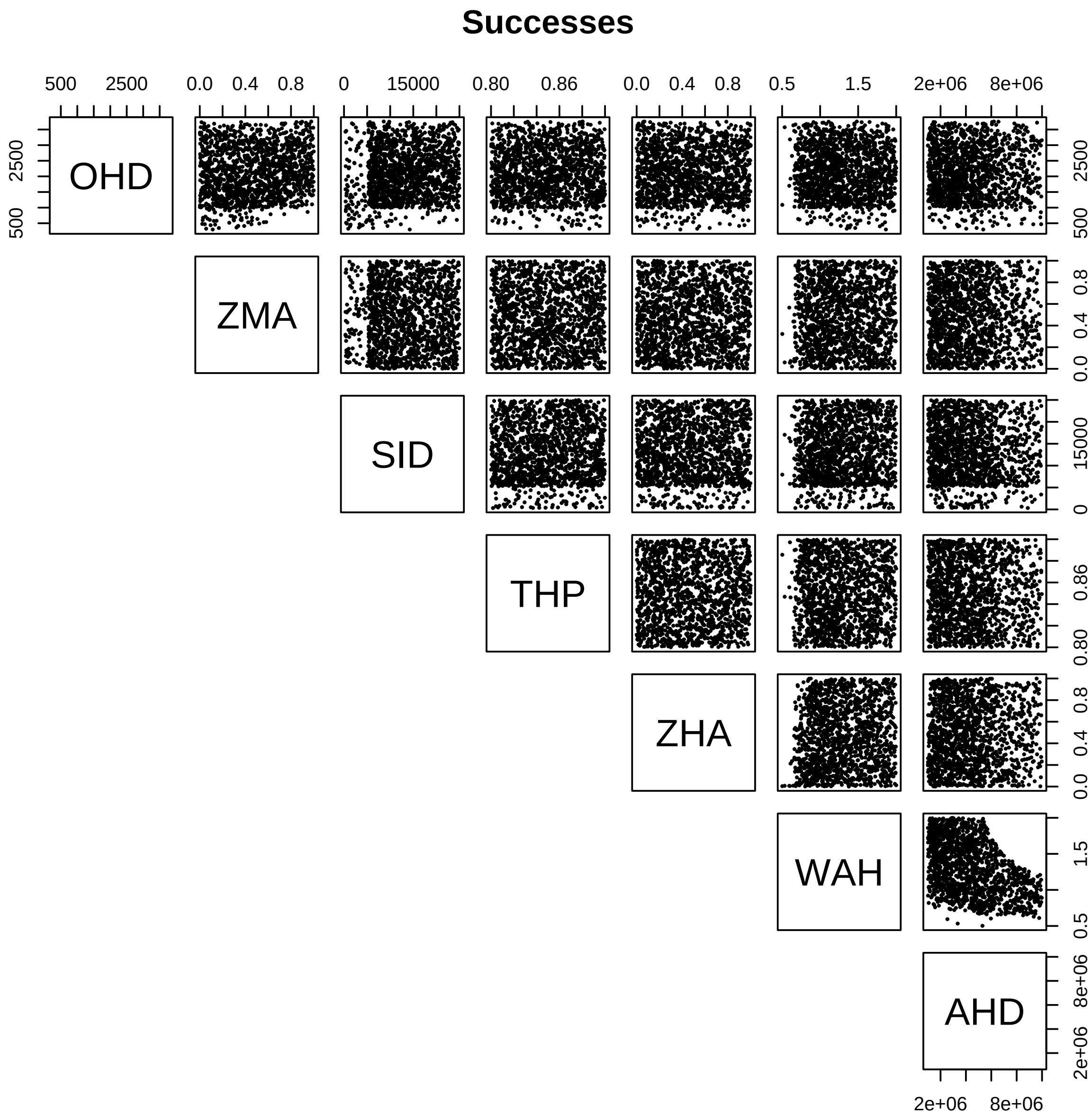
  false $+$ves Accepting an x that will fail;
  false $-$ves Rejecting an x that will succeed.

- Post-analysis of our actual evaluations can help us:

Outcome	Predicted, $\nu = 50\%$			Predicted, $\nu = 0.5\%$		
	Fail	Succ.	Err. rate	Fail	Succ.	Err. rate
Fail	781	241	23.6%	375	647	63.3%
Succ.	125	1353	8.5%	4	1474	0.3%


Fails
OHD
ZMA
SID
THP
ZHA
WAH
AHD
500
2500
0.0
0.4
0.8
0
15000
0.80
0.86
0.5
1.5
2e+06
8e+06

[Back]


Successes
OHD
ZMA
SID
THP
ZHA
WAH
AHD
500
2500
0.0
0.4
0.8
0
15000
0.80
0.86
0.5
1.5
2e+06
8e+06

[Back]