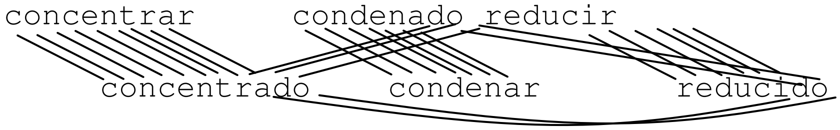
Processing units vs. prosodic units

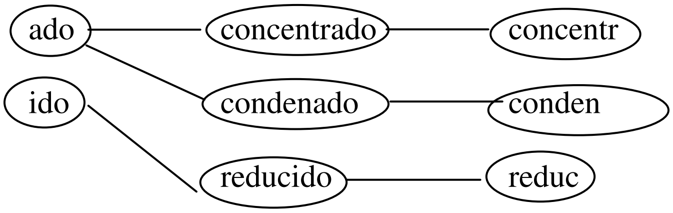
(1) First, a loose end from Bybee: Morpheme independence—Spanish past participles

Bybee argues that *–ado* can't be stored as a separate morpheme, because its behavior depends on the frequency of the word in which it appears. But that follows only if storage as a separate morpheme precludes whole-word storage. In Hay's model, we saw that the two coexist.

Bybee's model of lexical representation



Hay's model



(If the bases are bound forms, then what affects their resting activation is not their frequency in isolation, but only, as with affixes, how often they get accessed in derived words.)

It's hard/wrong to have intuitions about these models, as always—we need to write down equations or run simulations[1]—but here's how I *think* the predictions differ.

Bybee: Frequent words get more reduced. Reduction of *–ado* in one word spreads to other *–ado*s. Why is *–ado* more reduced than *–ido*? Either the words that *–ado* occurs in tend to be frequent (or, it just occurs in a higher number of frequent words, assuming no inhibitory effect from the infrequent words it occurs in), or it's an effect of the difference in deletion rate after *a* vs. *i*.

⇒ More reduction in frequent words; more reduction in affixes that occur in lots of frequent words.

Hay: Frequently-accessed strings get more reduced. Because the *ado* and *ido* strings are contained within the suffix (don't cross a morpheme boundary—at least under the implied morphological analysis above), whole-word entries for participles that are accessed frequently get more reduced, and entries for suffixes that are accessed frequently. (There could also be a phonetic effect of *a* vs. *i*.)

⇒ More reduction in words that are more frequent than their bases (that's just a rough measure—really, the increased reduction should be in words that are whole-accessed more frequently than their bases are accessed); more reduction in affixes that occur in lots of words that are less frequent than their bases (same caveat).

[1] For a discussion and illustration of why relying on intuition is bad—and what to do instead—see Partha Niyogi & Robert Berwick (1997). Populations of learners: the case of Portuguese. *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society.*

The data on p. 152 are consistent with either theory. If Hay's model is right, then the difference between high- and low-frequency words is coming from the fact that more of the high-frequency words are more frequent than their bases.

(2) Prosodic constituency vs. Bybeean processing units
Bybee proposes that "words that are often used together become processing units" (p. 157) and this leads to "phonological fusion". This is pretty much the Hay-ian view of what happens within words.

[Of course, we need to define 'often used together' (or let the model define it implicitly)—do we just mean that the sequence is frequent, or that it's more frequent than would be expected given the frequencies of the components and some assumptions about how things combine?]

How might prosodic units be different from processing units?

(3) I. grammar-dependence vs. distribution/usage dependence
Here are some possible worlds...

(i) Units determined entirely by the grammar
E.g., ALIGN(LxWd,L,PWd,L) (⇒ compounds 2 words, prefixes and proclitics left out, suffixes and enclitics folded in); p-word then acts as rule domain

(ii) Units determined entirely by processing
Sequences stored as units (or, accessed in unit stored form, even if decomposed alternatives exist) display phonological fusion internally, propensity to alternate at edges.

In general, (i) should predict a cleaner pattern than (ii), with fewer frequency effects on individual items. (i) also predicts tidy interaction with (presumably) non-processing considerations such as prosodic minimality.

(iii) Processing masquerading as grammar
Say that processing privileges left edges in such a way that prefixes and proclitics are, in general, more likely than suffixes and enclitics to get left out of the processing unit. If the tendency is strong enough, it could look like the ALIGN constraint above, perhaps with some lexical exceptions.

Similarly, effects of affix length, and differences between compounding and affixation (a given morpheme presumably participates in a wider variety of compounds than it does affixed forms) could come out of a processing model. If strong enough, they could look like grammar (plus exceptions).

(iv) Grammar with processing-grounded constraints
We often appeal to phonetic motivations for constraint rankings—why not appeal to a processing motivation for the tendency ALIGN(LxWd,L,PWd,L) >> ALIGN(LxWd,R,PWd,R)?

We need a processing version of something like Hayes's "inductive grounding"[2] (which deals with how messy and fine-grained phonetic patterns could get phonologized into coarse phonological constraints).

(v) Grammar that can refer to processing
This is something I've tried—constraints like ALIGN(AccessedUnit,L,PWd,L). The idea is to be able to generate a cleaner pattern than the pure processing story would predict, by letting the grammar run things, in the usual way, with limited opportunities for processing to have its say.

But of course, you want to see if your processing model can generate the clean-looking pattern on its own, à la (iii)—see below.

(4) II. hierarchical structure
One of the main ideas in Nespor & Vogel and early Selkirk was that different rules would refer to different domains. E.g. primary stress (and associated lack of vowel reduction) at the p-word level, stress retraction at the p-phrase level, spirantization at the intonational phrase level.

What could be the equivalent of prosodic levels in the processing approach? Perhaps looser processing units. In Bybee's English example (6.4.1), maybe English *don't* is very strong, *I don't* less so, *I don't know* still less, and *I don't know him* maybe pretty weak (though not as weak as *the production of linguistic material*).

This should predict that rule applicability is actually gradient and not tied to well-defined domains. E.g. stress retraction is a weaker rule, that applies only to tightly cohering units, and spirantization is stronger, applying even to more loosely cohering units.

- Let's discuss some predictions that this would make.

- What do you think about primary stress assignment in a framework like this (take the Italian case, where there can be a clear difference between primary and secondary stress, because of vowel reduction)—how do we make it obligatory that every stem or (say) disyllabic prefix has to get a primary stress?

(5) III. large units
Once we get up to units like the intonational phrase and utterance, it's implausible that we're dealing with stored units very often. Many intonational phrases will never have been heard or used by the speaker before. But phonology is nonetheless sensitive to those units (or so it's claimed).

- How could a processing theory deal with large units? Does it make any different predictions that the prosodic theory, and are they plausible?

An attempt at simulation...

[2] Bruce Hayes (1999). Phonetically-Driven Phonology: The Role of Optimality Theory and Inductive Grounding. In Michael Darnell, Edith Moravcsik, Michael Noonan, Frederick Newmeyer, and Kathleen Wheatly (eds.), *Functionalism and Formalism in Linguistics, Volume I: General Papers*. Amsterdam: John Benjamins. Pp. 243-285.

(6) Baayen et al.'s[3] MATCHECK model

Given a lexicon (morphemes and morpheme combinations), predict which parse will win.

Dutch Example (BS p. 1281)

bestelauto	*be+stel+auto*	('delivery van')	possible and most likely to come to mind
	bes+tel+auto	('berry counting car')	possible but unlikely to come to mind
	bes+t+el+auto	--	real morphemes but illegal combination

MATCHECK models the timecourse of word recognition.

- When a lexical entry reaches an activation threshold (actually, when its share of the total activation, $p(w, t)$, reaches a threshold), it is copied into a memory buffer.
- When a set of lexical entries that span the target word (e.g. *be*, *stel*, *auto*) exists in the buffer, that parse of the target becomes available.
- First full parse to become available is assumed to have "won" (but we could also assign interpretations to other information, such as how long it took the first parse to become available, and which other parses become available when).

(B&S p. 1286)

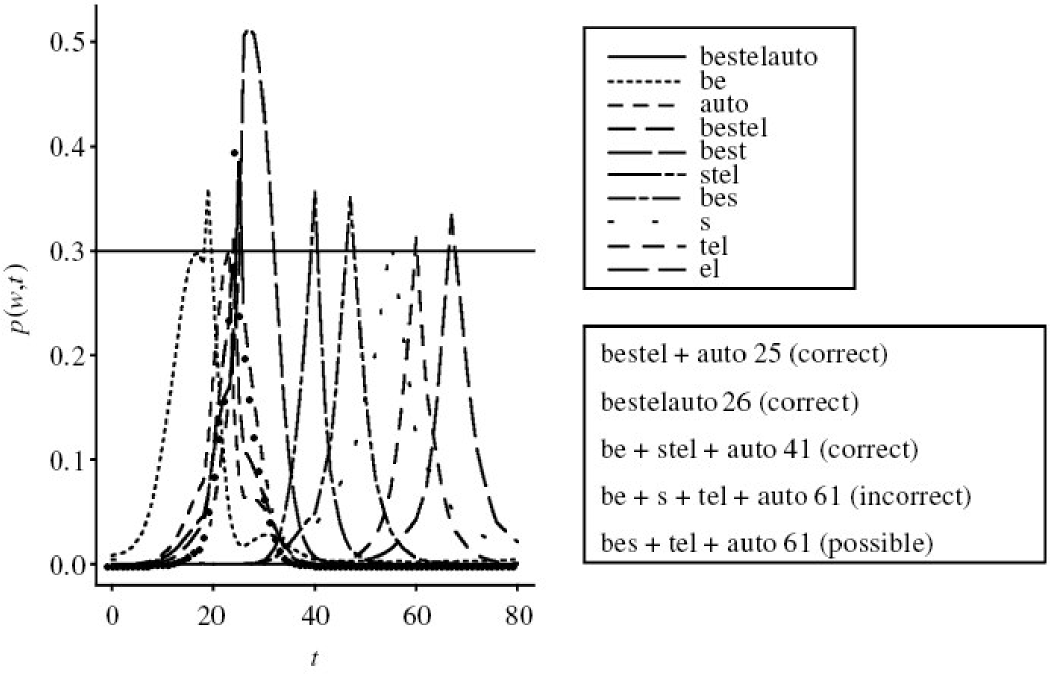


Figure 2. Probability of identification $p(w, t)$ for selected access representations as a function of time-step t, with activation threshold $\theta = 0.3$, for *bestelauto*, 'delivery van'. The time-steps at which full spannings become available are listed in the lower right-hand corner.

[3] Harald Baayen, Robert Schreuder, and Richard Sproat (2000). Modeling morphological segmentation in a parallel dual route framework for visual word recognition. In Frank van Eynde & David Gibbon (eds.) *Lexicon Development for Speech and Language Processing*. Pp. 267-293.
Harald Baayen & Robert Schreuder (2000). Towards a psycholinguistic computational model for morphological parsing. *Transactions of the Royal SocietyLondon* A 358: 1281-1293.

(7) What determines activation?
If an entry's weight is still being allowed to increase, the preceding activation is just multiplied by that node's decay rate:

$a(w, t) = a(w,t\text{-}1)/\delta_w$ (*a* stands for 'activation'; *w* identifies the entry; *t* is the current timestep; δ_w is the decay rate for that node)

If the entry's weight is determined to have peaked, the activation asymptotes out to the original (resting) activation:

$a(w, t) = |a(w,t\text{-}1)\text{-}a(w,0)|*\delta_w$

What determines when an entry starts decreasing its activation?

If an entry has not yet reached threshold, and it is either edge-aligned with the target* or similar enough† to the target, then it gets to keep increasing.

(*or, edge-aligned with a substring of the target that can be formed by stripping off outer affixes that have reached threshold; e.g., *in* is edge-aligned with *uninformed* if *un* has already reached threshold)
(†similarity is length of the target, if entry is superstring of target; otherwise, similarity is length of the target minus Levenshtein edit distance between entry and target; entry is "similar enough" if its similarity ≥ *t*)

What determines an entry's decay rate? Informally, it's a combination of its length, its resting activation, and how much of the target it matches.

$$\delta_w = \delta\frac{L(w)}{L(w)+\frac{\alpha}{L(w)}\log(a(w,0))} + \left(1-\delta\frac{L(w)}{L(w)+\frac{\alpha}{L(w)}\log(a(w,0))}\right)\left(\frac{|L(w)-L(T)|}{\max(L(w),L(T))}\right)^{\zeta}$$

if $\zeta > 0$, and otherwise $\delta_w = \delta$

where L(*i*) is the length of *i*, *T* is the target word, and there are free parameters $\alpha > 0$ (spike parameter), $\zeta > 0$ (forest parameter), and $0<\delta<1$ (overall decay rate).

(8) Testing the model's predictions—example from the literature
Hay & Baayen (2002)[4] find that, for a set of Dutch words in *-heid* Matcheck's parsing times are correlated with subjects' lexical-decision reaction times (from a previous study).

[4] Jennifer Hay & Harald Baayen (2002). Parsing and productivity. In Geert Booij & Jap van Marle (eds.) *Yearbook of Morphology 2001.* Kluwer Academic Publishers. Pp. 203-235.

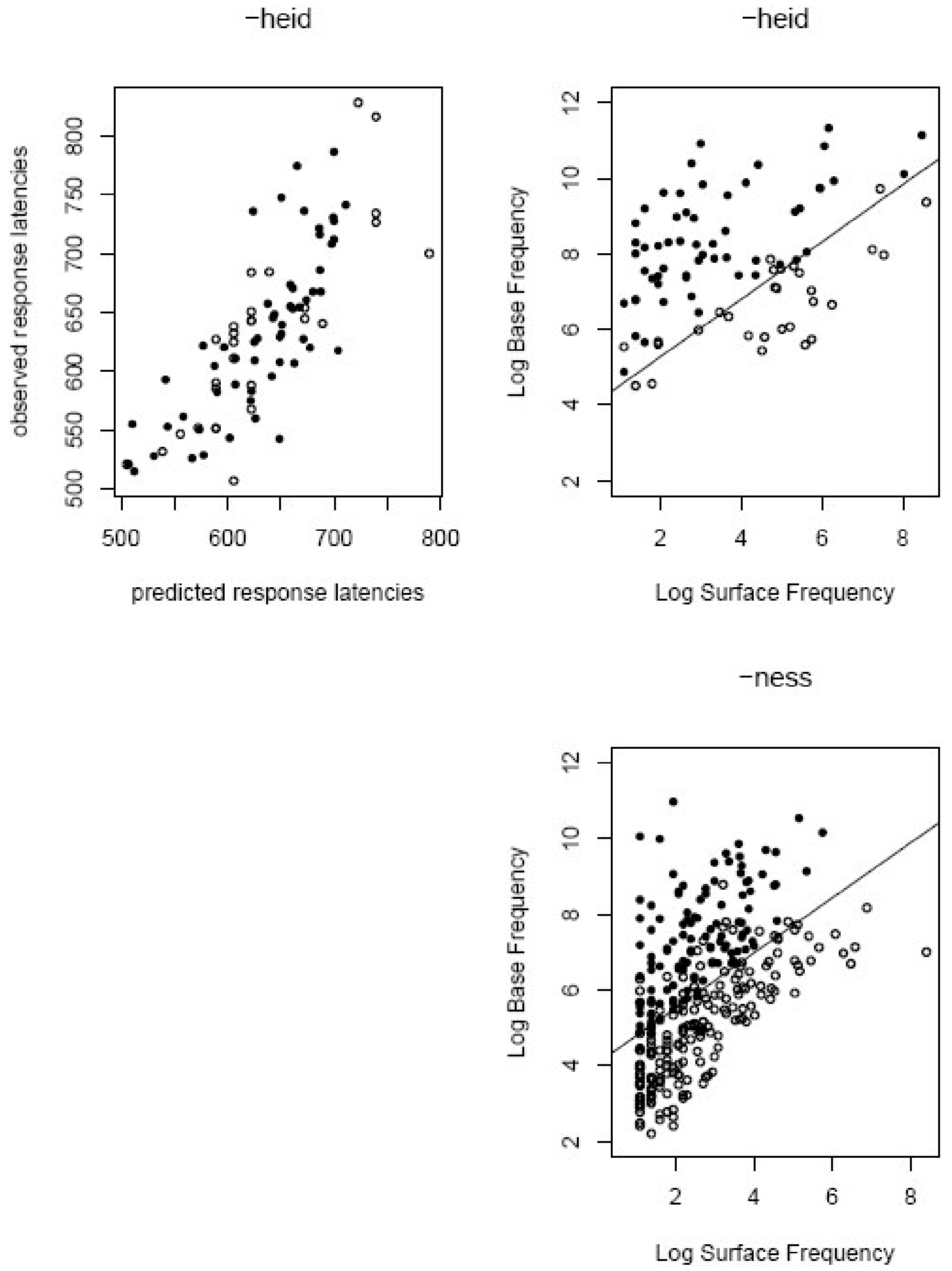


Figure 2: Left Panel: The correlation between the model times produced by MATCHECK and observed response latencies for the Dutch suffix *-heid*. Upper Right Panel: The relation between log derived frequency and log base frequency for *-heid*. Bottom Right Panel: The relation between log derived frequency and log base frequency for *-ness*. Solid points represent forms for which the parsing route is the first to produce a complete spanning in MATCHECK. Open points represent forms for which the derived form is the first to provide a complete spanning. The lines in the right panels optimally divide the words that are parsed from those that are recognized on the basis of their derived forms.

If we split up the words...

- For words where the whole-word parse is faster, only the time to the whole-word parse is significantly correlated with RT.

- For words where the decomposed parse is faster, the time to the whole-word parse and the morphological family size are sig. correlated with RT.

➢ Why does a large family speed RT in the decomposedly-parsed items?

Assume that, on top of the form-similarity relations implicit in Matcheck, there are explicit connections between words that contain the same morpheme, and they spread activation to each other. Thus, the stem gets activated faster when it occurs in lots of other words.

"Now consider the case in which the parsing route wins the race. The present experimental data on *-heid* suggest that in this case activation spreads into the morphological family. This makes sense, as initially the comprehension system knows only that it is dealing with a stem that has to be combined with some affix. By allowing the morphological family members to become co-activated, all and only the possibly relevant candidates are made more available for the processes which combine the stem with the derivational suffix and which compute the meaning of their combination.
"In fact, since the derived form is one of the family members of the base, it will be activated more quickly when the base has a large morphological family. This is because it is embedded in a larger network of morphologically related words, and the morphologically related words spread activation to each other. This may explain why log derived frequency remains such a strong predictor of the response latencies even for the words in the P set [i.e., the words where the decomposed parse wins]." (pp. 13-14 of ms. version)

➢ Why doesn't family size matter for whole-word-parsed items?

This part of the paper is fuzzy to me. I'll just quote:

"Consider what happens when the direct route is the first to provide a complete spanning of the target word, say, *snel-heid*, "quickness", i.e., "speed". Once the derived form has become available, the corresponding meaning is activated, apparently without activation spreading into the morphological family of its base, "snel". In other words, family members such as *ver-snel-en* ("to increase speed") and *snel-weg* ("freeway") are not co-activated when *snel-heid* has been recognized by means of the direct route." (p. 13 of ms. version)

It's hard to reconcile this with the idea that words containing the same stem are connected. Ideas?

➢ Why doesn't the time to the decomposed parse matter?

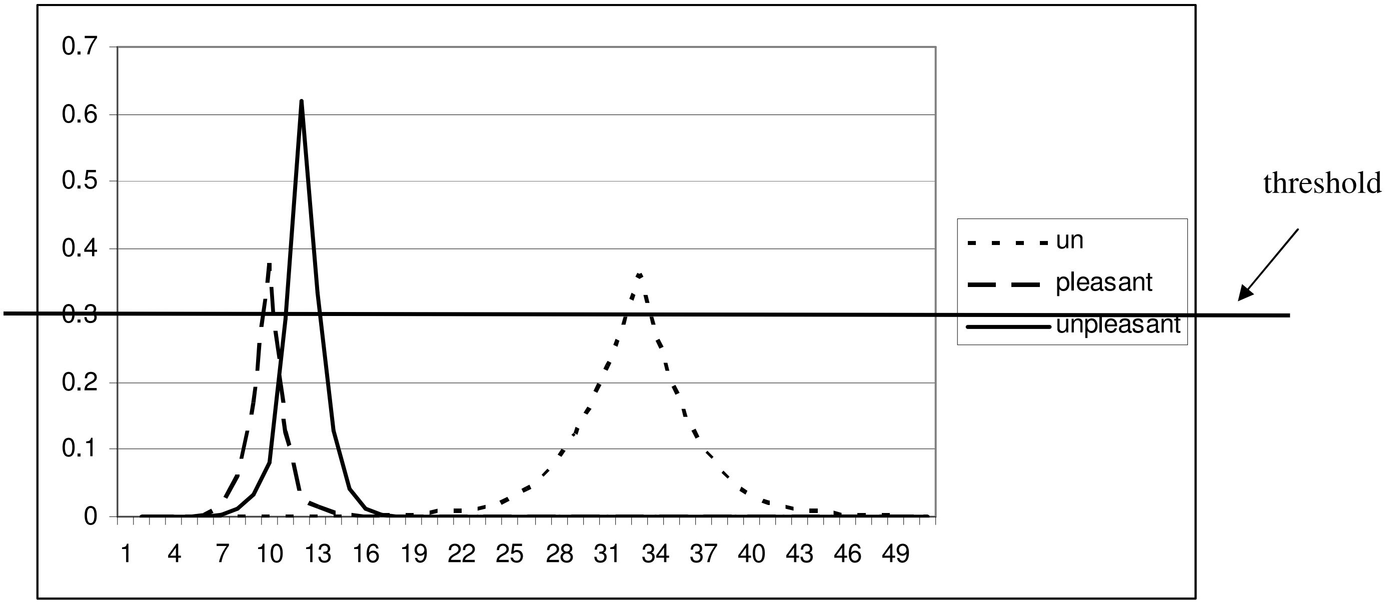
"We think that derived frequency and family size may conspire to mask an effect of the timestep itself at which the base word itself becomes available, leading to the absence of a measurable effect of base frequency and t_P [the time at which the decomposed parse becomes available].” (p. 14 of ms.)

(9) Testing the model’s predictions w.r.t. prosodic structure
I implemented Matcheck on Monday/Tuesday with the intention of testing all of Raffelsiefen’s English examples and seeing whether whole-word access, as predicted by the model, correlates with simple p-word structure, as diagnosed by Raffelsiefen.

Then I wanted to look at compounds—do they tend strongly to get a complex structure, simply because they’re infrequent compared to their component stems? Etc., etc.

Problem is, my program is so slow that I’ve only managed to run one word, *unpleasant*! Here are the results:

unpleasant becomes available at *t*=11
un+pleasant becomes available at *t*=32



Is that bad news for predicting “prosodic structure” from lexical access mode? Not necessarily. For frequency data, I used Adam Kilgariff’s BNC frequency lists (http://www.kilgarriff.co.uk/bnc-readme.html). Specifically, I used the unlemmatized ‘demographic’ (conversation) subset, restricted to types with token frequency > 5 (yields 10,596 form-unique types).

All resting activations were simply the token frequencies in this file. That means that the resting activation of *un-* is just 29 (I don’t know what these would have been—transcriptions of hesitation noises?) which is surely too low:

...

12	ulster
13	ultimate
12	ultra
29	um
29	umbrella
110	umm
29	**un**
6	'un
7	unable
60	unbelievable
154	uncle
9	unclear
11	uncles
33	uncomfortable
10	unconscious
834	under

...

In B&S, BSS, affix resting activation is the sum of the frequencies of all the words that contain the affix! So, most likely *un+pleasant* would have won if I'd done it that way. (But I didn't want to, because it would mean examining words by hand.)

(10) What's the point of all this?

The goal is to test all the *...V/d/V...* words of Tagalog (prediction: those that are parsed decomposedly, with a boundary on one side or the other of the *d*, will not undergo intervocalic tapping). But clearly I need to make my code faster first.

Then, the real goal—Kevin and I will work on this, if all goes well—is to make the resting activations dynamic and simulate exposure to a whole lexicon (Kevin's idea: in the form of running text). If *un+pleasant* wins, then *un* and *pleasant* get a frequency boost; by if *unpleasant* wins, then it gets a frequency boost.

(I will post the code (perl) on the class web page in case you'd like to play with it, on the condition that you all promise not to mock it! Suggestions for improvement, especially in speed, will be welcome though.)

Next time (nothing to read): I want to talk about Tagalog and/or Palauan, maybe with some simulation results if I can speed up the code. I'll send e-mail about upcoming readings...