# Protection and vulnerability level

Thanks for the reminder to be specific! Updated below.

Complex biogenic structures mediate the process of predation on reef habitats. To account for this, we reduce the rate at which a predator encounters prey at a level proportional to the availability of appropriately sized **predation refuges**, which prevent prey fish from being encountered. The refuge function, $R_j(w_p)$, describes the proportion of fish of size $w_p$ in prey group $j$ that are hidden from predators. We refer to the set of proportions that describe refuge availability across the entire size range of model fish as a **refuge profile**. $1 - R_j(w_p)$ is then the proportion of fish, of weight $w_p$ in prey group $j$ that are vulnerable to consumption by predators.

Individuals smaller than $w_{settle}$ represent pelagic larval fish that have not yet settled to the reef substrate and are too small to utilize refuge to avoid predation. The default value for this minimum weight is 0.1 grams. All individuals below this size are assumed to be vulnerable to predators. To ensure some food is always available, the maximum value of $R_j(w_p)$ is set to $r_{max}$, which defaults to 98%.

Refuge can be suppressed for functional groups which are not expected to utilize biogenic structures for protection (e.g. schooling species) with the [refuge_user] parameter in the species parameters data frame.

The mizerReef package provides three methods to define the overall refuge profile of a reef depending on the availability of complexity data for the modelled area.

**Simple Method** This method is best suited to data-poor reefs or reefs where the refuge distribution is unknown. The proportion of prey of weight $w_p$ and functional group $j$ with access to refuge $R_j(w_p)$ is given by:

$$R_j(w_p) = \frac{r}{1 + e^{(\alpha(w - W_{refuge}))}}$$

{#eq-refuge_simple}

where $r$ defines the proportion of fish smaller than size $W_{refuge}$ with access to predation refuge. The availability of refuge decreases with increasing body size at rate $\alpha$. A smaller $\alpha$ results in a more gradual decrease in refuge availability, while the default value of 100 results in a steeper decline.

**Binned Method** This method is appropriate for theoretical applications and does not rely on complexity data. It sets the availability of refuge to a constant proportion of fish within a given size range. The proportion of fish in functional group $j$ and weight $w_p$ with access to refuge $R_j(w_p)$ is given by:

$$R_j(w_p) = r_k \qquad w_p \in (\ w_{k-1}, w_k\ ]$$

where $w_p$ is the weight of the prey and $r_k$ is the proportion of fish with access to refuge in size class $k$.

**Data Method** This method is appropriate when **refuge density** data is available for the modeled reef. The **refuge density** describes the distribution of refuges across defined fish body size categories. The proportion of prey of weight $w_p$ and functional group $j$ with access to refuge $R_j(w_p)$ is given by

$$R_j(w_p) = \tau \cdot \frac{\eta_k}{\sum_i \int_{w_{k-1}}^{w_k} N_i(w)\ dw} \qquad w_p \in (\ w_{k-1}, w_k\ ]$$

when there are fewer refuges than competitors and $r_{max}$ otherwise. The parameter $\tau$ is the proportion of fish with access to refuge that are expected to utilize it, $\eta_k$ is the density of refuges in size range $(w_{k-1}, w_k]$ and $\sum_i \int_{w_{k-1}}^{w_k} N_i(w)\ dw$ gives the total number of fish from any group in size range $(w_{k-1}, w_k]$. This represents the density of competitors for refuges in size class $k$.

**Vulnerability** The vulnerability of prey of weight $w_p$ in group $j$ to predation by a predator in functional group $i$ is then given by:

$$V_{ij}(w_p) = 1 - \nu_i \cdot R_j(w_p)$$

where $\nu_i$ represents whether predators in functional group $i$ are able to access prey within refuge. For functional groups where $\nu_i$ is false, some aspect of their morphology or foraging strategy allows them to access prey hidden in refuge; for example large eels that can fit into crevices much smaller than their body size. For these predators, vulnerability for all prey is set to 1.

### Changes to feeding level

In mizerReef models, feeding level only applies to herbivorous and detritivorous functional groups. Predation is regulated by refuge, and no maximum intake rate is imposed for piscivores.

For all non-piscivorous functional groups, encountered food is consumed subject to a standard Holling functional response type II to represent satiation. This determines the *feeding level* $f_i(w)$, which is a dimensionless number between 0 (no food) and 1 (fully satiated) so that $1 - f_i(w)$ is the proportion of the encountered food that is consumed.

## QUESTION

I wanted to include feeding level for non-piscivores that are feeding on plankton, and in the consumption of detritus and algae. To implement this in mizer, I created a new species parameter `pisc` which is true for piscivores and false for non-carnivorous species (which will also be helpful for plotting!)

The feeding level matrix is still calculated as normal, but then I set the feeding level to 0 for all piscivores, as follows. Will this cause issues? Could I also achieve this by setting h to inf for piscivores in the species params matrix?

```r
feed <- encounter / (encounter + params@intake_max)
# Set feeding level to 0 for all piscivores
idx.pisc <- which(as.logical(params@species_params$pisc))
feed[idx.pisc, ] <- 0
```

### Changes to Encounter Rate

To account for prey vulnerability, the encounter rate $E_i(w)$ is modified to

$$E_i(w) = \gamma_i(w) \int \sum_j \theta_{ij} V_{ij}(w_p) N_j(w_p) \phi_i(w, w_p) w_p \, dw_p + E_{A.i}(w) + E_{D.i}(w).$$

where $V_{ij}(w_p)$ sets the vulnerability of prey species $j$ and weight $w_p$ to predation by predator species $i$. Vulnerability is determined by the presence of predation refuge.

## QUESTION

I am just looking for some confirmation that the way I've implemented this makes sense and does what I think its doing.

In the code, vulnerability information is kept as two separate arrays:

- `vulnerable` – equivalent to $1 - R_j(w_p)$, an array prey species x prey size, gives proportion vulnerable to being encountered at each size
  - In `reefEncounter` this matrix is multiplied by `n` to obtain `n_vulnerable`, a prey size x prey species array giving the number of fish at each size that are vulnerable to being encountered/consumed by predators
- `bad_pred` – equivalent to $\nu_i$, a vector of Boolean values stored in `species_params` that indicates whether refuge should impact predator group foraging or not
  - In `reefEncounter`, the rows of the encounter matrix (predator species) are filled in according to `bad_pred`. When `bad_pred` is TRUE, `n_vulnerable` is calculations instead of `n`.

---

### Changes to predation mortality

All consumption by fish translates into corresponding predation mortalities on the ingested prey individuals. $V_{ij}(w_p)$ is the proportion of prey species $i$ and weight $w_p$ that are encountered by a predator of group $j$. The rate at which all predators from group $j$ consume prey of size $w_p$ is

$$\texttt{pred\_rate}_j(w_p) = \int \phi_j(w, w_p)(1 - f_j(w))\, \gamma_j(w) N_j(w)\, dw.$$

The mortality rate due to predation is then obtained as

$$\mu_{p.i}(w_p) = \sum_j \texttt{pred\_rate}_j(w_p)\, V_{ij}(w_p)\, \theta_{ji}.$$

**QUESTION** I think I finally understand where you are coming from that vulnerability needs to be in `predMort`. To make this happen, I do the following:

```
+ create a list containing the prey species by prey size vulnerability matrix for each predator species
+ loop through the list, returning pred_mort = pred_mort + Vul_j * pred_rate
```

which should return the total predation mortality by prey species and prey size from all predators. I guess what I'm concerned about is, is this going to kill the same number of fish that are encountered? In other words, I'm worried that multiplying encounter rate by V(w) isn't the same as multiplying predMort by V(w). I'm probably just confused!

Code is below, feel free to ignore it and shout 'YOURE ASKING TO MUCH OF ME!!!'

```
reefPredMort <- function(params, n, n_pp, n_other, t, pred_rate,
                         vulnerable = reefVulnerable(params, n, n_pp,
                                                     n_other, t), ...) {
# Find indices of fish that have grown out of the resource spectrum
    idx_sp <- (length(params@w_full) -
                   length(params@w) + 1):length(params@w_full)
    no_sp <- nrow(params@species_params)

    # Find indices of predator species whose foraging is hindered by refuge
    bad_pred  <- which(params@species_params$bad_pred == TRUE)
    good_pred <- which(params@species_params$bad_pred == FALSE)

    # Create list of vulnerabilities for each predator
    vul <- vector("list", no_sp)
    vul[bad_pred] <- list(vulnerable)
    vul[good_pred] <- list(1)

    # Loop through predator species to calculate predation mortality on
    # each prey species by predator
    pm <- vector("list", no_sp)
    for (i in 1:no_sp){
        pm[[i]] <- pm + vul[[i]] * pred_rate[, idx_sp, drop = FALSE]
    }

    return(base::t(params@interaction) %*% pm)
```

which should return the total predation mortality by prey species and prey size from all predators. I guess what I'm concerned about is, is this going to kill the same number of fish that are encountered? In other words, I'm worried that multiplying encounter rate by V(w) isn't the same as multiplying predMort by V(w). I'm probably just confused!

**QUESTION:** I also am having an issue where some of functions should be throwing up error messages, but aren't and I'm not sure why. For example, setRefuge() gives an error when a's or b's are missing from the species parameter dataframe, but when I ran newReefParams(), which calls setRefuge, on a set of params without a's and b's, I didn't get an error. I am happy to dig into this one on my own, but I thought I'd mention it in case it was something simple!