test_OMELETmodel.stan

`./OMELET/OMELET_rstan/model059_0h4h/test_OMELETmodel.stan` を例に、stan scriptの説明。

Overall structure

stanの文法では、ブロックと呼ばれる項目に分けて記述を行うことになっている。

- `functions` ブロック (lines 1-165)：自作関数を記述。
- `data` ブロック (lines 167-209)：入力データを定義。R側から渡すデータをこちらで定義。
- `parameters` ブロック (lines 211-225)：推定するパラメータ。こちらに書かれているパラメータについてMCMCで事後分布が推定される。
- `transformed parameters` ブロック (lines 227-248)：`parameters` ブロックで定義されているパラメータに何らかの処理を行った後の値を定義。
- `model` ブロック (lines 250-294)：尤度や事前分布の式を記述する。`data` や `parameters`、`transformed parameters` ブロックで定義された変数を利用する。
- `generated quantities` ブロック (lines 295-327)：サンプリングで得られたパラメータを利用してさらに何かの計算を行う場合にその計算を記述。

data

`data` は `test_OMELETmodel_mkdata.R` に定義されている `mkdata_now` functionで作られる。入力データは、Rからstanにlist形式で渡す必要がある。

- `int<lower=1> num_r;`：代謝ネットワークの反応数 r'
- `int<lower=1> num_ri`：独立フラックス u_l の数 (number of independent fluxes) ${\rm rank}({S})$
- `int<lower=1> num_rd`：従属フラックス ${v}_j^d$ の数 (number of dependent fluxes) f
- `int<lower=1> num_rc`：尤度計算をする反応数 (number for calculation) r
- `int<lower=1> num_m`：代謝物数 (number of metabolites) m
- `int<lower=1> num_mc`：endでない、つまりネットワークの端に位置しない代謝物。この代謝物の周りについて定常からの逸脱を考慮することでフラックスの分散共分散行列を指定する (number of metabolites for calculation)
- `int<lower=1> num_b`：生成物のelasticity coefficients `b` の数
- `int<lower=1> num_ri_wt`：基準となる条件における独立フラックスの数。植松論文ではWT fasting stateが基準となる条件で、G6PC fluxを1に固定する。そのため独立フラックスの数が他の条件より1減ることになる。
- `int<lower=1> num_smpl`：全ての条件を合わせた時のサンプル数 (number of samples)
- `int<lower=1> num_g`：条件数 (number of groups) g
- `int<lower=1> num_p`：酵素量の予測を行う反応数 (number of predictions)。これは転写物量データのみ得られていて酵素量データが得られていない反応について、転写物量を用いて酵素量を推定する反応を意味する。植松論文ではGPTとGLUDの2反応。

- `int<lower=0> num_met_eff;`: effectors (アロステリック制御因子やcofactors、2つ以上基質・生成物がある場合の2つ目以降)の数
- `int<lower=0> num_met_eff_pairs;`: `met_eff_list_include.txt` に記載された、全ての「代謝物-反応名-制御の様式（アロステリック制御かcofactorか）」のペアの数
- `int<lower=0> num_enz_cmplx;`: 複合体を形成する酵素タンパク質の数
- `int<lower=0> num_met_est;`: 量をパラメータとして予測する代謝物の数
- `matrix[num_m,num_r] S`：化学量論行列 S
- `matrix[num_r,num_ri] N`：
- `matrix[num_r,num_ri] Nu`：W^u（昔の変数名の名残 N for u）
- `matrix[num_r,num_rd] Ne`：$W^{\dot{x}}$（昔の変数名の名残 N for errors）
- `matrix[num_mc,num_r] Sp`：S^{+} (S plus)
- `matrix[num_mc,num_r] Sm`：S^{-} (S minus)
- `int idx_calc[num_rc]`：尤度計算に用いる反応のindex (index for calculation)。
- `int idx_g[num_g,2]`：各サンプルがどの条件に属するかを示すindexの始まりと終わり (index for groups)。例えば [1, 11] だとサンプルの1番目から11番目までが同じ条件に属することを意味する。
- `int idx_p[num_p]`：酵素量の予測を行う反応のindex (index for prediction)
- `int idx_p_[num_rc-num_p]`：酵素量の予測を行わない反応のindex
- `int idx_b[num_b-1]`：生成物のelasticity coefficientsを評価するindex。長さが `num_b-1` になっているのは確かGLUDは可逆だが生成物であるalpha-ketoglutarateが計測できていないため生成物のelasticity coefficientsを推定しないようにしているため。
- `int is_indflux[num_r];`: 独立フラックスに相当する反応なら1、そうでなければ0のlogical配列
- `int idx_indflux[num_ri];`: 独立フラックスのインデックス
- `int idx_include[num_rc];`: 尤度計算に用いる反応のインデックス
- `int is_irrev[num_rc]`：可逆 0(reversible) もしくは不可逆 (irreversible) を表すlogical配列。
- `int idx_irrev[num_rc-num_b];`: 不荷逆反応のインデックス
- `int is_cmplx[num_rc];`: 複合体を形成する反応のインデックス
- `vector[num_smpl] enz[num_rc]`：酵素量データ (enzyme) e
- `vector[num_smpl] rna[num_rc]`：転写物量データ (RNA)
- `vector[num_smpl] sub[num_rc]`：基質量データ (substrate)。これは x の一部。
- `vector[num_smpl] pro[num_rc]`：生成物量データ (product)。これは x の一部。
- `real<lower=0,upper=1> c_v`：フラックスのばらつきを決める固定パラメータ c^u
- `real<lower=0,upper=1> c_e`：フラックスの定常状態からの逸脱を決める固定パラメータ $c^{\dot{x}}$
- `vector[num_smpl] met_eff[num_rc];`：Effectors (cofactors and allosteric effectors) の代謝物量データ (metabolites as effectors)。植松論文では、1"NAD+" 2"NADH" 3"ATP" 4"ADP" 5"GDP"" 6"GTP" 7"FAD" 8"AMP" 9"Cit" 10"F1,6P" 11"Ala" 12"AcCoA" 13"Phe" 14"Leu"
- `int v_max`：フラックスがとりうる範囲。固定パラメータ。

parameters

- `vector[num_rc] v[num_g]`：フラックス v
- `real<lower=0> a[num_rc]`：基質のelasticity coefficients
- `real<lower=0> b[num_b-1]`：生成物のelasticity coefficients
- `vector<lower=-v_max,upper=v_max>[num_ri_wt] mu_vi_wt_tmp`：基準となる条件における独立フラックスの一部。G6PC fluxの値が1になるように調節するためにこのように一部だけパラメータとして指定している。

- `vector<lower=-v_max,upper=v_max>[num_ri] mu_vi_g[num_g-1]`：基準となる条件以外の条件における独立フラックス。
- `real<lower=0> sigma_n`：酵素量の尤度を評価する際の誤差項 $\sigma^{\hat e}$
- `real<lower=0> sigma_p`：mRNA-to-protein ratioの事前分布の分散 σ^{β}
- `real<lower=0> sigma_n2`：転写物量の尤度を評価する際の誤差項 $\sigma^{\hat t}$
- `vector[num_rc] r_p[num_g-1]`：ある1条件以外の条件におけるmRNA-to-protein ratio。
- `vector[num_rc-num_p] r_p_tmp`：ある1条件におけるmRNA-to-protein ratio。酵素量を転写物量から推定する反応では、mRNA-to-protein ratioと転写物量から酵素量を推定するが、酵素量の全条件での平均は1なので、自由度が1減る。つまりその条件におけるmRNA-to-protein ratioは、他の条件におけるmRNA-to-protein ratioと転写物量から求められる。よってある1条件におけるmRNA-to-protein ratioのみ別のパラメータで指定している。この条件は最後のindexの条件としている。
- `real<lower=0> e_effs[num_met_eff_pairs];`：アロステリック制御因子やcofactorといった基質・生成物以外のelasticity coefficients
- `vector<lower=0,upper=4>[num_g-1] met_est[num_met_est];`: 推定する代謝物量

transformed parameters

このブロックでは以下の3つの値を計算している。

- `vector[num_smpl] x[num_rc]`Equation (10) の $1/(v_j^0(1+{\epsilon}*j*^{*top*} *ln {x}*{kl}))$ に相当する部分。
- `vector<lower=0>[num_smpl] y[num_rc]`：その他の酵素について、Equation (13) の $1/\beta_{jl} \hat{e}_{jkl}$に相当する部分。

これらの値は `functions` ブロックで定義した関数を用いて計算されているので、そちらで詳細を記述する。

model

Likelihoods

lines 260-269のfor文では、各反応 `r` について、酵素量と転写物量の尤度を計算する。

```
for(r in 1:num_rc){
      if(r==idx_p[1]){
        target += normal_lpdf(rna[r] | y[r], sqrt(sigma_n2) );
      }else if(r==idx_p[2]){
        target += normal_lpdf(rna[r] | y[r], sqrt(sigma_n2) );
      }else {
        target += normal_lpdf(enz[r] | v_x[r], sqrt(sigma_n) );
        target += normal_lpdf(rna[r] | y[r], sqrt(sigma_n2));
      }
  }
```

尤度について、

- `target += normal_lpdf(rna[r] | y[r], sqrt(sigma_n2));`：転写物量の尤度。Equation (13) の$\mathcal{N}(t_{jkl} | \hat{t}_{jkl},(\sigma^{\hat{t}})^2)$ に相当。

- `target += normal_lpdf(enz[r] | v_x[r], sqrt(sigma_n) );`：酵素量の尤度。Equation (10) の $\mathcal{N}(e_{jkl} | \hat{e}_{jkl},(\sigma^{\hat{e}})^2)$ に相当。

If文について、

- `if(r==idx_p[1])` と `else if(r==idx_p[2])`：酵素量が転写物量から推定されている場合、転写物量の尤度のみを計算する。
- その他：酵素量と転写物量の尤度を計算する。

Priors (flux)

lines 281-285では、代謝フラックスの事前分布を指定する。

```
mu_vi_ref = prep_mu_vi_ref(mu_vi_wt_tmp,num_ri,Nu);
   target +=
calc_log_prior(mu_vi_ref,Nu,Ne,Sp,Sm,c_v,c_e,num_r,num_m,num_ri,num_mc,idx_
calc,num_rc,v[1]);
  for(g in 2:num_g){
    target += calc_log_prior(mu_vi_g[g-
1],Nu,Ne,Sp,Sm,c_v,c_e,num_r,num_m,num_ri,num_mc,idx_calc,num_rc,v[g]);
  }
```

`calc_log_prior` という関数で計算しているのが、Equation (6)の$\mathcal{N}({v}_l | {\mu}_l^v, {\Sigma}_l^v)$ と Equation (3)の $\mathcal{N}({u}_l | {\mu}_l^u, {\Sigma}_l^u)$ の積に相当。stan model中で u は変数として定義していない。

Priors (mRNA-to-protein ratio)

lines 271-279 ではmRNA-to-protein ratioの事前分布を指定する。

```
for(g in 1:num_g){
    if(g != num_g){
      target += normal_lpdf(r_p[g]|0,sigma_p);
    }else{
      for(i in 1:num_p){
        target += normal_lpdf(r_p_tmp[i]|0,sigma_p);
      }
    }
  }
```

いずれも、Equation (14) の $\mathcal{N}(\beta_{jl} | 1, (\sigma_l^{\beta})^2)$ に相当。stan model中で `normal_lpdf` の平均が0になっているのは昔のバージョンの名残で、後述する `prep_y` 中のline 159で1を足しているので論文中の式と実質的に同じになる。

Priors (elasticity coefficients)

いずれのelasticity coefficientsの事前分布も半正規分布。Equation (11) の $\mathcal{H}(\epsilon_{ji}|1)$ などに相当。ただしstan model中では全て正の値とし、反応速度式の中で符号をつけているため、全て同じ事

前分布として書いている。

```
target += normal_lpdf(a|0,1);
target += normal_lpdf(b|0,1);
target += normal_lpdf(e_effs|0,1);
```

Priors for error terms

誤差項の事前分布として、半コーシー分布を置いている。

```
target += cauchy_lpdf(sigma_p|0,0.5);
target += cauchy_lpdf(sigma_n|0,0.5);
target += cauchy_lpdf(sigma_n2|0,0.5);
```

functions

一部わかりにくい記述があるかもしれないが、stanの記法がいまいち融通が効かず、また高速化のためベクトル化をはかりstan内での演算を最小限にしようとした結果、今の形に落ち着いた。

calc_mu

Equation (6) 中の ${\mu}_l^v = {W^u}{\mu^u}_l$ に相当。

calc_Signa2

Equation (6) 中の ${\Sigma^v}_l = {W^u} {\Sigma^u}_l {W^u} + {W^{\dot{x}}}{\Sigma^{\dot{x}}}_l {W^{\dot{x}}}$ に相当。

calc_log_prior

Equation (6)の$\mathcal{N}({v}_l | {\mu}_l^v, {\Sigma}_l^v)$ と Equation (3)の $\mathcal{N}({u}_l | {\mu}_l^u, {\Sigma}_l^u)$ の積に相当。

calc_x

Equation (10) の $1/(v_j^0(1+{\epsilon}*j^{\top} \ln {x}*{kl}))$ に相当する部分を各反応について計算する。

例えばPgm2の場合、

```
x[1] = ones ./(mean_v[1]*(1+a[1]*sub[1]-b[1]*pro[1]));
```

このstan model全体に言えることだが、高速化のためベクトル化を行っている。つまり1つ1つ数値を計算するのではなく、ベクトル単位での計算を行っている。

`x[1]` というのは line 39 で定義されているように、`num_s` （`num_smpl`と同じ）の長さをもつベクトル。

`ones` は長さが `num_s` で要素が全て1のベクトル。

`./` は要素単位の演算を表す。例えば `[4 6] ./ [2 3]` は `[2 2]` となる。

`mean_v` というのは Equation (10) の v_j^0 に相当。

`(1+a[1]*sub[1]-b[1]*pro[1]))` は Equation (10) の $(1+{\epsilon}_j^{\top} \ln {x}{kl})$ に相当。
elasticity coefficientsをw基質、生成物（、cofactor、allosteric effectors）に分けてパラメータとして設定しているため、それぞれについて代謝物との積を計算している。ただしこの代謝物量はデータを入力する時に既に対数変換を行っているので再度変換は不要。

prep_mu_vi_ref

基準となる条件でのG6PC fluxの事前分布の平均が1となるようにフラックスの事前分布を決める。

prep_mean_v

Equation (10) の v_j^0 を計算する。つまり全条件でのフラックスの事前分布の平均。

prep_r_p_all

mRNA-to-protein ratioを準備する。前述のように、ある1条件におけるmRNA-to-protein ratio。酵素量を転写物量から推定する反応では、mRNA-to-protein ratioと転写物量から酵素量を推定するが、酵素量の全条件での平均は1なので、自由度が1減る。つまりその条件におけるmRNA-to-protein ratioは、他の条件におけるmRNA-to-protein ratioと転写物量から求められる。よってある1条件におけるmRNA-to-protein ratioのみ別のパラメータで指定している。この作業を行う。

prep_y

Equation (13) の $\hat{t}_{jkl}$ を計算する。

`v_x[r][idx_g[g,1]:idx_g[g,2]]` が $\hat{e}_{jl}$ に相当。

`1+r_p_all[g][r]` が β_{jl} に相当。

prep_met_est

量をパラメータとして推定する代謝物（植松論文ではoxaloacetate (OAA) と pyruvate (Pyr) ）の相対的な代謝物量をパラメータで指定する。

他の代謝物と同様、OAAとPyrの代謝物量の全条件での平均は1になるので、自由度が1減る。つまりある条件におけるOAAとPyrの相対的な代謝物量は、他の条件におけるOAAとPyrの相対的な代謝物量から求められる。この作業を行っている。

ただし他の代謝物も対数変換して入力としているので、OAAとPyrも対数変換した値を推定することになっている。

generated quantitites

このブロックはサンプリングに影響しないが、サンプリング評価のために以下を計算する。

- `vector[num_smpl] log_lik[num_rc]`：酵素量と転写物量の対数尤度。WAICの計算に用いるため。
- `real log_prior[num_g]`：事前分布の確率。
- `vector[num_smpl] enz_pred[num_rc]`：posterior predictive checkに用いるための、事後分布を用いてサンプリングした酵素量。
- `vector[num_smpl] rna_pred[num_rc]`：posterior predictive checkに用いるための、事後分布を用いてサンプリングした転写物量。