1. Comparison of simulations after 25 rounds with simulations after 50 rounds.

As remarked in the paper, in actual practice, e.g., in wetlabs, usually no more than 20 or 25 total rounds of positive and negative SELEX are carried out. In our computations, we took 50 – 80 total rounds to be sure that we were near the asymptotic limits. However, inspection of the Figures S1, S2, S3 with Figures 2, 3, 4 of the paper and the figure captions below, shows that, in most cases, we were fairly close to these limits after 25 rounds.

(a) Here $[T_v] = 1 \mu M$ is fixed for each round.

(b) Initially $[T_v] = 1 \mu M$ and decreases like $1/r$.

**Figure S1.** Concentration of nucleic acid fractions after 25 rounds for negative SELEX only. $[NA] = 1 \mu M$. We see that the same conclusion deduced from Figure 2 in the paper, holds here also. Namely, it is better fix the target in going from round to round in negative selection rather than to reduce it. (In positive SELEX, the contrary is true.)

*Date: February 3, 2014.*
(a) Here \([T_\nu]\) = 1\,\mu M is fixed for each round.

(b) Initially \([T_\nu]\) = 1\,\mu M and decreases like \(1/r\).

Figure S2. Efficiencies for negative SELEX after 25 rounds. \([NA]\) = 1\,\mu M. Notice that although the efficiencies in panel (b) are much higher than in panel (a), we do not get the poorest binder by decreasing the negative target from round to round. (Compare with panels (a,b) in Figure S1.) The final efficiency in Figure S1, panel (a) after 25 rounds is \(E_\nu^{(25)} = 0.2830\) whereas after 50 rounds in Figure 3, panel (a) it is \(E_\nu^{(50)} = 0.2847\). The final efficiency in Figure S1, panel (b) after 25 rounds is \(E_\nu^{(25)} = 0.9935\) whereas after 50 rounds in Figure 3, panel (b) it is \(E_\nu^{(50)} = 0.9966\).
The fifth target component was absent during the negative rounds of the simulations. The nucleic acid pool size was reset between rounds (PCR). In panel (a), after 25 rounds,}\]
\[
\hat{\mathbf{F}}^{(25)} = (F_8, F_9, F_{10}, F_{12}, F_{16}) = (0.2039, 0.183, 0.094, 0.390, 0.1263),
\]
whereas in the paper (first 50 rounds shown in Figure 4, panel (a)), the “limiting” NA fraction vector \( \hat{\mathbf{F}}^{(150)} = (F_8, F_9, F_{10}, F_{12}, F_{16}) = (0.1956, 0.2794, 0.0498, 0.384, 0.0908) \). Notice that in the former case, \( \sum_{i \in \{8, 9, 10, 12, 16\}} F_i^{(25)} = 0.9978 \) whereas in the latter case, \( \sum_{i \in \{8, 9, 10, 12, 16\}} F_i^{(150)} = 1.000 \) to four significant figures. Thus essentially most of the nucleic acids to be selected have been selected after 25 rounds.

For panels (b) and (c) we alternated positive selection with negative selection, beginning with positive selection. In panel (b), for \( \lambda = 0.6 \), After 25 rounds, \( \hat{\mathbf{F}}^{(25)} = (F_8, F_{10}, F_{12}, F_{16}) = (0.4833, 0.1316, 0.1603, 0.2243) \), and the “limiting” NA fraction vector \( \hat{\mathbf{F}}^{(150)} = (F_8, F_{10}, F_{12}, F_{16}) = (0.3754, 0.1496, 0.2581, 0.2169) \) at the end of negative selection. Again \( \sum_{i \in \{8, 10, 12, 16\}} F_i^{(25)} = 0.9994 \) whereas \( \sum_{i \in \{8, 10, 12, 16\}} F_i^{(150)} = 1.000 \) to four significant figures. While the concentration fractions are somewhat different, the inequalities \( F_8 > F_{16} > F_{12} > F_{10} \) hold.

For panel (c) with \( \lambda = 0.4 \), after 25 rounds, \( \hat{\mathbf{F}}^{(25)} = (F_{17}, F_{20}) = (0.7115, 0.2803) \), while in Figure 4, panel (c), the “limiting” NA fraction vector \( \hat{\mathbf{F}}^{(150)} = (F_{17}, F_{20}) = (0.7915, 0.2085) \). Again the total fraction after 25 rounds is within one percent of unity.

For panel (d) with \( \lambda = 0.0 \), after 25 rounds, \( \hat{\mathbf{F}}^{(25)} = F_{13} = 0.9454 \), while in Figure 4, panel (d), the “limiting” NA fraction vector \( \hat{\mathbf{F}}^{(150)} = F_{13} = 1.000 \). The total fraction after 25 rounds is within six percent of unity.
2. MATLAB code for alternate SELEX: Positive SELEX followed by negative SELEX

Alternate.m

Input data for this program consists of positive integers $M, N$, an $M \times N$ matrix of positive entries, the affinity matrix $A$, an $M$ vector, the initial target vector $T_s^{(0)} = (T_{s,1}^{(0)}, \ldots, T_{s,M}^{(0)})$, the nucleic acid pool size for positive selection, $[NA]$, the index $k$ corresponding to the removed target component $T_{ν,k}$ for negative SELEX and a multiplier $m$ that leaves the proportion of positive to negative rounds unchanged.

One example of $3 \times 5$ affinity matrix with entries (in $1/μM$) in the physical range is given below:

\[
\begin{array}{cccccc}
A(1,1) &=& 4629.6; & A(1,2) &=& 5492.0; & A(1,3) = 1623.4; & A(1,4) = 3386.5; & A(1,5) = 4420.2; \\
A(2,1) &=& 3164.6; & A(2,2) &=& 792.0; & A(2,3) = 1938.0; & A(2,4) = 4800.0; & A(2,5) = 1925.5; \\
A(3,1) &=& 1091.7; & A(3,2) &=& 793.0; & A(3,3) = 4630.0; & A(3,4) = 2445.0; & A(3,5) = 2103.8;
\end{array}
\]

The reader should be aware of a notational change, namely the replacement of subscripts “ν” in the paper by “n” in the executable code because Matlab does not permit the use of the former subscript there.

% Alternate SELEX iteration scheme
function Alternate(M, N, A, NApos, Ts0, delete, Rs0, Rn0, m)
format long;
Rs=Rs0*m; % Rs is the number of positive selection steps, $R_s$, and $m$ is a multiplier.
Rn=Rn0*m; % Rn is the number of negative selection steps, $R_ν$.

% The selection fraction $λ = Rs0/(Rs0 + Rn0)$ where the positive integers Rs0, Rn0 have no common
% divisor larger than one.
% Here NApos = [NA], the fixed pool size for positive SELEX steps.

% “delete” is a fixed positive integer, $1 \leq delete \leq M$ that denotes the deleted target component in
% negative SELEX. k=10; % The number of iterations of alternate SELEX.

% In each round, Rs rounds of positive selection and Rn rounds of negative selection are performed.

% Randomly selected initial NA fractions, $\hat{F}^{(1)}$.
F=rand(1,N);
F=F/sum(F); % Normalization.
Fr(:,1)=F’;

% Initialization of the positive target vector, $T_s^{(0)} = Ts0$.
Ts(1,:)=Ts0;

for i=1:M
for j=1:N
K(i,j)=1/A(i,j); % $K_{ij}$ is the dissociation equilibrium constant for the $j^{th}$ NA species bound to
% the $i^{th}$ target species.
end
end

% We used the fixed point iteration scheme to solve for each of the free target components, $[Tf_i]$,
\( i = \{1, 2, \ldots, M\} \) for both positive selection and negative selection. This iteration scheme is less labor intensive than Newton’s method and converges almost as fast.

\% Main loop starts here.
for \( r=1:k \)

\% **Positive SELEX begins.** This subprogram uses the multiple target (positive) SELEX algorithm given in Section 4. It can be used as a stand alone program for positive SELEX.
for \( i=2:Rs \)

\( Ts(i,:)=(1-1/i)*Ts(i-1,:); \)
% Infinite positive target dilution, \( T_{s}^{(r+1)} = (1 - s_r)T_{s}^{(r)} \), where \( s_r = 1/r \).
end

\( Fs(:,1)=Fr(:,r); \)
% This will be the starting NA fractions for positive SELEX updated after one or several rounds of negative selection.

\% Positive selection ends.
\( Ts(1,:)=Ts(Rs,:); \)
\% Negative SELEX begins.
\% This subprogram uses the negative SELEX algorithm in Section 5. It can be used as a stand alone program for negative SELEX whether or not a subtarget is deleted. Here the subprogram is written for one deleted target component.
\( Mneg=3; \)
% Negative target vector \( Tn = T_{n} \) with the size of total target concentration of \( Tn = T_{n} = 100 \mu M \).
\( Tn(1,:)=Ts0*100; \)
for j = 1:M
if j == delete
Tn(1,j)=0;
else
Tn(1,j)=Ts0(1,j);
end
end
Tn(1,:)=100*Tn(1,:)/sum(Tn(1,:));
for i=2:Rn
Tn(i,:)=Tn(1,:); % Total negative target concentration is fixed from round to round.
end
NA(r,1)=1; % Starting nucleic acid concentration, [NA], for negative SELEX.
Fn(:,1)=Fs(:,Rs+1); % Set the updated NA fractions after positive selection as the initial NA fractions for negative selection. (Here \( \hat{F}_n = \hat{F}_{\nu} \))
for q=1:Rn
freeTn=FixedPoint(Tn(1,:),K,Fn(:,q),NA(r,q),Mneg,N,A); % Fixed point iteration for \( \overrightarrow{T_{\nu f}} \).
for i=1:Mneg
Tnf(q,i)=freeTn(i); % Free target vector \( \overrightarrow{T_{\nu f}} \) resulting from fixed point iteration.
end
for i=1:Mneg
omegan(q,i)=Tnf(q,i)/sum(Tnf(q,:)); %Free target fraction vector \( \overrightarrow{\omega_{\nu}} \) normalized from \( \overrightarrow{T_{\nu f}} \).
end
for j=1:N
t=0;
for i=1:Mneg
t=t+Tnf(q,i)*A(i,j);
end
Dn(j,q)=t; % Compute \( D_{\nu,j} = \sum_{i=1}^{M} [T_{\nu f}l]A_{ij} \).
end
den=0;
for l=1:N
den=den+Fn(l,q)/(1+Dn(l,q)); % Compute \( [NAf] = [NA] \sum_{j=1}^{N} \frac{F_{\nu,j}}{1+D_{\nu,j}} \). Here we take \([NA] = 1 \mu M\).
end
for j=1:N
Fn(j,q+1)=(Fn(j,q)/(1+Dn(j,q)))/den; % Compute updated nucleic acids fractions, \( F'_{\nu,j} \).
end
NA(r,q+1)=NA(r,1); % \([NA]\) is fixed from round to round for PCR amplification.
end
% Negative selection ends.
Fr(:,r+1)=Fn(:,Rn+1); % Updated NA fractions after \( R_\nu \) rounds of negative selection.
end

FixedPoint.m
% Fixed point iteration method.
function Tf=FixedPoint(T,K,F,NA,M,N,A)
format long;
for i=1:M
tf(1,i)=0; % Start with an initial guess as a zero vector.
end
for r=1:10 ^\,(20) % Number of iterations for fixed point.
for j=1:N
s(j)=0;
end
for j=1:N
for l=1:M
s(j)=s(j)+tf(r,l)*A(l,j); % Compute $D_{j,f}$.
end
end
for i=1:M
u=0;
for j=1:N
u=u+F(j)*A(i,j)/(1+s(j)); % Compute $\sum_{j=1}^{N} \frac{F_jA_{ij}}{1+D_{j,f}}$.
end
tf(r+1,i)=T(i)*(1+NA*u) \,(−1);
end
% Compute the free target vector $[Tf^{(k+1)}]$, where $k$ is the iteration number for fixed point iteration.

if norm(tf(r+1,:)-tf(r,:),inf) < 10\,(−15)
m=r+1;
Tf=tf(m,:);
break
else
Tf=tf(r+1,:);
end
% A tolerance of $10^{-15}$ for convergence of fixed point iteration was used here. The iteration will be repeated until the relative error becomes smaller than $10^{-15}$.