11-3-2010

Computational Biology

Harvey Greenberg
University of Colorado Denver

Allen Holder
Rose-Hulman Institute of Technology, holder@rose-hulman.edu

Computational Biology

H. Greenberg, A. Holder

Mathematical Sciences Technical Report Series
MSTR 10-09

November 3, 2010

Department of Mathematics
Rose-Hulman Institute of Technology
http://www.rose-hulman.edu/math

Fax (812)-877-8333 **Phone (812)-877-8193**

COMPUTATIONAL BIOLOGY

Harvey J. Greenberg
University of Colorado Denver

Allen G. Holder
Rose-Hulman Institute of Technology

Computational biology is an interdisciplinary field that applies the techniques of computer science, applied mathematics, and statistics to address biological questions. OR is also interdisciplinary and applies the same mathematical and computational sciences, but to decision-making problems. Both focus on developing mathematical models and designing algorithms to solve them. Models in computational biology vary in their biological domain and can range from the interactions of genes and proteins to the relationships among organisms and species.

Genes are stretches of deoxyribonucleic acid (DNA), which is sometimes called the “User Manual for Life" and is a double-stranded helix of nucleic acids bonded by base-pairs of complements (a-t, c-g). The *central dogma* of molecular biology asserts that information in a cell flows from DNA to ribonucleic acid (RNA) to protein (note, Francis Crick used ‘dogma’ when he introduced this in 1958 to mean ‘without foundation’ because there was no experimental evidence at that time). Proteins are the “workers” of the cell, and there is much focus on recognizing, predicting, and comparing their properties.

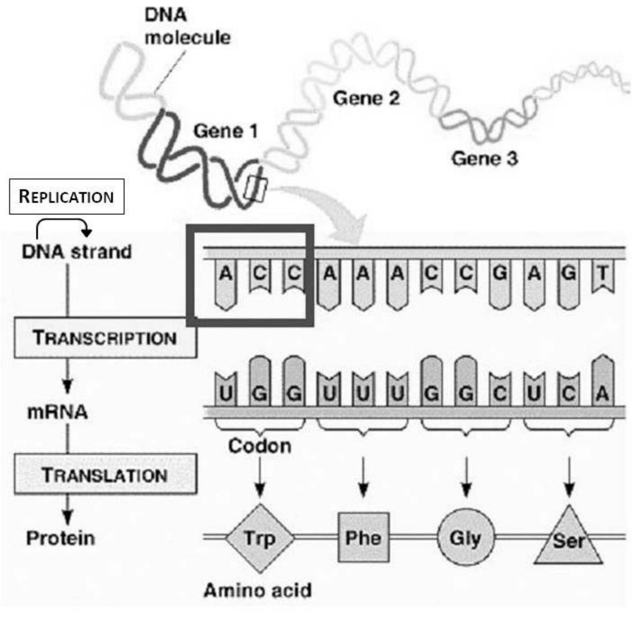


Figure 1: Central Dogma of Molecular Biology

Proteins interact either directly by modifying each other’s properties through direct contact or indirectly by participating in the production and modification of cellular metabolites. Collectively, the biochemical reactions and the possible intermediates that produce a metabolite are called a metabolic pathway, and a metabolic network is a collection of these pathways. The study of complex networks like that of the metabolism is called systems biology.

Linear Programming: A linear program (LP) is an optimization problem in which the variables are in $\mathbb{R}^n$, and the constraints and the objective are linear.

Flux Balance Analysis (FBA) — A biochemical process is defined by n reactions that convert m compounds:

$$a_{1j}x_1 + \cdots + a_{mj}x_m \underset{k_-^j}{\overset{k_+^j}{\rightleftharpoons}} b_{1j}x_1 + \cdots + b_{mj}x_m,$$

where x_i is the concentration of the i^{th} compound, and $k_\pm^j$ is the j^{th} reaction rate (for a 2-way reaction the reverse rate need not equal the forward rate). The corresponding ODE is:

$$\frac{\mathrm{d}x_i(t)}{\mathrm{d}t} = \sum_{j=1}^n (b_{ij} - a_{ij}) \left(k_+^j x_1^{a_{1j}} \cdots x_m^{a_{mj}} - k_-^j x_1^{a_{1j}} \cdots x_m^{a_{mj}} \right) = \sum_{j=1}^n S_{ij} v_j(x),$$

where v is the flux (production or consumption of mass per unit area per unit time), and S_{ij} is defined as a "stoichiometric" (pronounced stoy-kee-uh-me'-trik) coefficient. These coefficients are interpreted as:

$$S_{ij} > 0 \Rightarrow \text{rate of compound } i \text{ produced in reaction } j;$$
$$S_{ij} < 0 \Rightarrow \text{rate of compound } i \text{ consumed in reaction } j.$$

The following holds asymptotically, provided that the system approaches a steady state toward equilibrium concentrations $\bar{x}$:

$$\lim_{t \to \infty} \frac{\mathrm{d}x(t)}{\mathrm{d}t} = Sv(\bar{x}) = 0. \tag{1}$$

Dropping the dependence of the flux on $\bar{x}$, the flux cone is defined by this homogeneous system plus non-negativity for one-way reactions, indexed by J:

$$\mathcal{F} = \{v : Sv = 0, \ v_J \geq 0\}. \tag{2}$$

In a metabolic network reactions are distinguished between external and internal. The flux associated with an external reaction is an exchange between the network of interest and the cell's environment.

$R_1:$	$A \rightharpoonup B + 2C$	(multiple output)
$R_2:$	$C + 2D \rightharpoonup B$	(multiple input)
$R_3:$	$2B \rightharpoonup D$	(simple)
$R_4:$	$2C \leftrightharpoons 3D$	(simple, 2-way)
$E_1:$	$\rightharpoonup A$	(supply)
$E_2:$	$B \leftrightharpoons$	(2-way exchange)
$E_3:$	$D \rightharpoonup$	(demand)

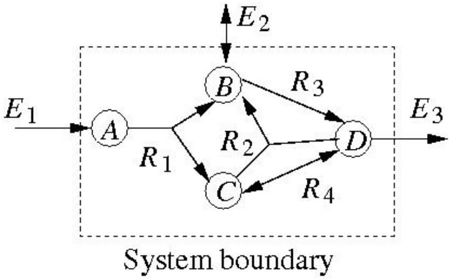


Figure 2: Example metabolic network with four internal and three external reactions.

The stoichiometric matrix for the internal reactions is extended to include external reactions, each being a singleton column with ±1:

$$S = \begin{array}{cccc|ccc|c}
R_1 & R_2 & R_3 & R_4 & E_1 & E_2 & E_3 & \\
-1 & 0 & 0 & 0 & 1 & 0 & 0 & A \\
1 & 1 & -2 & 0 & 0 & -1 & 0 & B \\
2 & -1 & 0 & -2 & 0 & 0 & 0 & C \\
0 & -2 & 1 & 3 & 0 & 0 & -1 & D
\end{array}$$

All reactions are 1-way, except R_4 and E_2, so $J = \{1, 2, 3, 5, 7\}$, leaving v_4 and v_6 without sign restriction in the flux cone.

Strictly speaking a metabolic network is usually not a network in the OR sense because some internal reactions have multiple inputs or outputs (sometimes called a "process network" in chemical engineering). Hence, LP is used, rather than specialized network algorithms, to find fluxes. The FBA LP model has the form:

$$\max\ c^T v : \ \ v \in \mathcal{F} \cap \mathcal{B}, \tag{3}$$

where $\mathcal{B}$ is a bounding set so that the linear program has an optimal solution. A common objective is to maximize the rate of growth defined in terms of metabolites, where the objective coefficients (c) depend on the organism. Other objectives include maximizing some metabolite production, minimizing by-product production, minimizing substrate requirements, and minimizing mass nutrient uptake (Palsson, 2006).

An optimal basis depends on the definition of $\mathcal{B}$. Three possibilities, which may be combined, are:

simple bounds:	$L_K \le v_K \le U_K$
fixing inputs and/or outputs:	$v_K = \bar{v}_K$
normalization:	$\sum_{j \in K} v_j = b,$

where K is a subset of reactions. Inputs and outputs are generally a subset of the exchanges. Normalization applies to one-way reactions — i.e., $K \subseteq J$. Each extreme ray of the flux cone corresponds to an extreme point of the polytope. The converse is generally not true — viz., fixing the flux of a reaction that transports metabolites in or out of the cell can introduce extreme points with no extreme ray of the flux cone passing through them.

Pathways are subnetworks with a single biological effect. In an ordinary network, where each internal reaction has a single input and output, this is a path. A cut set is defined as a set of reactions whose removal renders the stoichiometric equation (1) infeasible for a specified output. For an ordinary network, the OR terminology is a disconnecting set. A minimal cut set for a specified output is, in OR terminology, simply a cut set. For the example, a cut set that separates D from the rest of the network is $\{R_1, R_3, R_4, E_1\}$. Finding a (minimal) cut set in the general case becomes an IP, using binary variables to block pathways to some specified output.

Nonlinear Programming: A nonlinear program (NLP) is defined by having the objective or some constraint function be nonlinear in the decision variables.

Protein folding — Most proteins go through a process that twists and turns the molecules from their primary state of a linear order of amino acids to a native three dimensional state in which it remains. That process is called "folding," and it is theoretically possible to predict a protein's native state, or structure, by knowing its primary state. This determines a protein's function, and some diseases (e.g., Alzheimer's, Huntington's, and cystic fibrosis) are associated with protein misfolding.

Predictive models became possible following the work of Christian B. Anfinsen, who in 1961 published experimental results supporting the Thermodynamic Hypothesis: *A protein's native state is uniquely determined by its primary sequence; it transitions to a state of minimum free energy.* This leads to a nonlinear program with the decision space defined as the spacial coordinates of atoms, constrained by the biochemistry of a protein's defining amino acid sequence. The objective function is a free energy determined by potential energies from atomic bonds and non-bond interactions.

The bonds for the sequence of amino acids shown in Figure 3 are covalent, meaning that they share electrons, and these strong bonds hold the backbone together. Objective terms for the i^{th} covalent bond include the energies required to stretch, bend, and twist the bond.

action	Energy	
stretching	E^{stretch}	$= \sum_i K_i^L (L_i - L_i^0)^2$
bending	E^{bend}	$= \sum_i K_i^\theta (\theta_i - \theta_i^0)^2$
twisting	E^{twist}	$= \sum_i K_i^\phi (1 - \cos(\omega_i))$

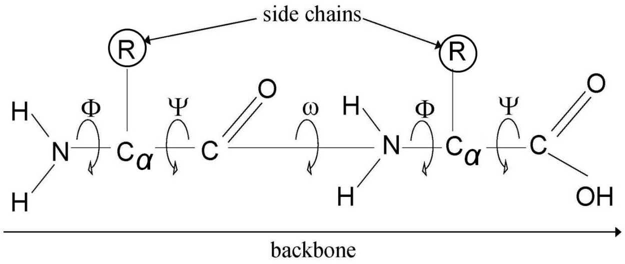


Figure 3: Covalent bonds along the backbone result in a residue for each of the amino acids. The torsion angles are denoted by Ψ and Φ; ω is the dihedral angle.

The variables are the bond length (L) and the bond angles, $\theta = (\Psi, \Phi)$ and ω, which are determined by atomic coordinates. Parameters include target values (L^0, θ^0). Weight parameters (K) are scale factors that put the energy terms in the same unit; those values can be measured or derived. For example, if it requires 100 kcal/mole to break a bond, and two positive charges within 3.3Å (Angstrom) have at least 100 kcal/mole, then total energy is reduced by breaking a bond to keep positive charges distant. Estimating these values to determine weight parameters is not an exact science, so even these basic energy functions are not exact, and there are other energy functions for non-covalent bonds and among non-bonding atoms.

Two common energy functions estimate the electrostatic and Van der Waals interactions:

action	Energy
Electrostatic	$E^{\text{elec}} = \sum_{i<j} K_{ij}^{\text{elec}} \frac{q_i q_j}{d_{ij}}$
Van der Waals	$E^{\text{vdw}} = \sum_{i<j} K_{ij}^{\text{vdw}} \left(\left(\frac{d_{ij}^*}{d_{ij}} \right)^{12} - \alpha_{ij} \left(\frac{d_{ij}^*}{d_{ij}} \right)^{6} \right)$

The variables are the pair-wise distances (d), which are determined by the atomic coordinates. Parameters are the atomic charges (q) and equilibrium distances (d^*).

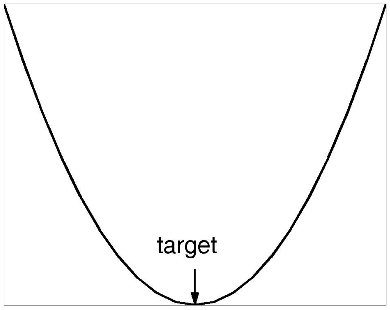


Figure 4: The squared deviation of E^{stretch} and E^{bend} is convex.

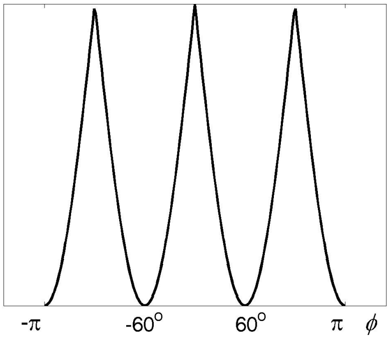


Figure 5: E^{twist} with $\omega = 3/2(\phi - \pi)$.

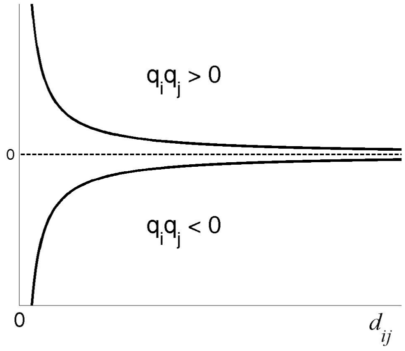


Figure 6: E^{elec} depends on the sign of q_iq_j. Oppositely-signed atoms attract, so the energy is negative and favors them being close.

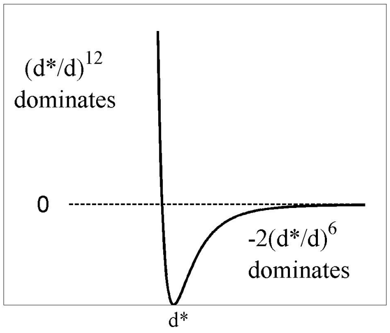


Figure 7: Lennard-Jones approximation of E^{vdw} for $\alpha = 2$.

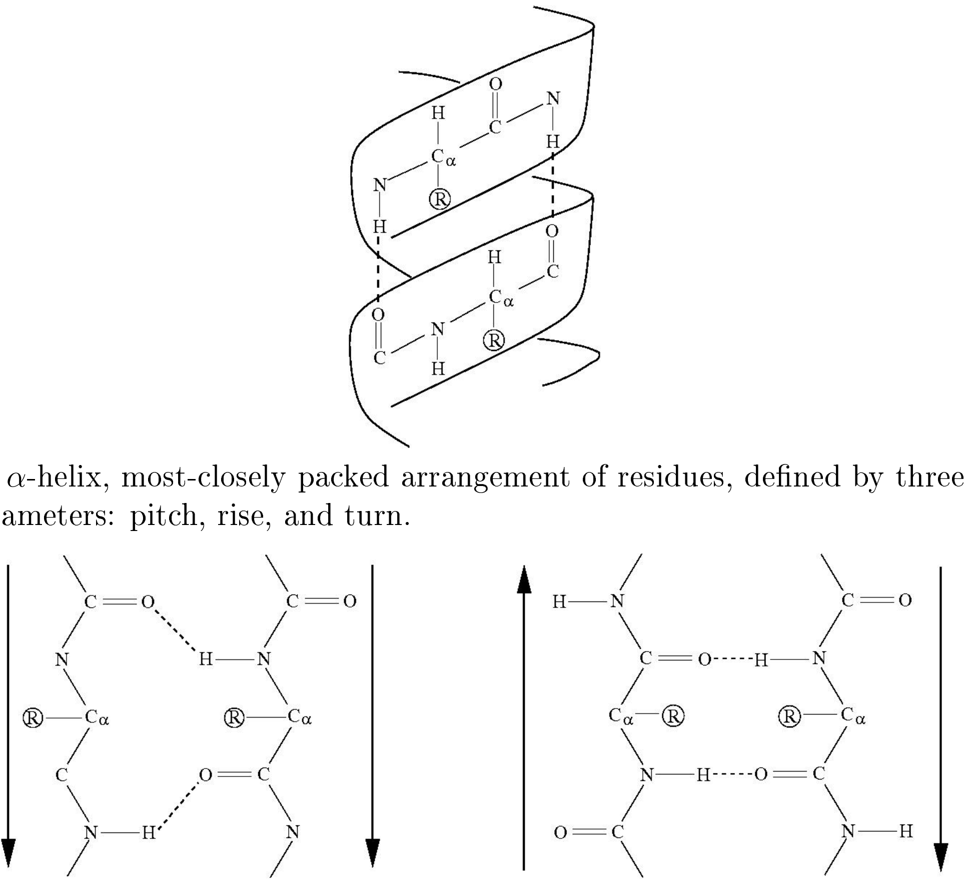
The NLP approach (Floudas and Pardalos, 2000) uses energy principles that underly molecular dynamics, and these methods attempt to find the native state and a pathway to it. In practice, not all parameters are grounded in some physical law. An energy function could include contributions from non-bonded and uncharged pairs, based on their distance and radii. Alternatively, known structures can be used to predict an unknown structure, based on their evolutionary similarity. This is called "homology," and it is focused on determining the native state and not on discerning the dynamic pathways to reach it.

The multi-modal shape of the energy landscape leads to the Levinthal Paradox: *many proteins reach their native state within milliseconds, yet the number of stable confor-*

mations grows exponentially in the number of amino acids. One explanation is that proteins fold into a nearby local minimum of the free energy instead of the global minimum. Global optimization methods based on this principle are called "funneling methods." Another explanation is that the dimension of the problem is not the length of the amino-acid sequence but is instead the number of chains that obey patterns not fully understood. Combinatorial optimization methods based on this principle are called "chain growth" and "zipping and assembly" algorithms.

Comparing Protein Function — A protein's function is determined by its 3D native state. The 3D confirmations of many proteins are known and are available from the Protein Database (www.pdb.org). Comparing protein structures relates protein function and collects proteins into functionally similar families that help identify a protein's functions.

Proteins typically have multiple functional domains, each of which would act as an independent protein if its amino acid sub-sequence had folded independently. Two proteins are considered to be functionally similar if they share a (nearly) common domain. Each domain is composed of secondary structures, notably α-helices and β-sheets, illustrated in Figure 8. In structure alignment the goal is to best align the secondary structures between two proteins' domains. The input to the alignment problem is a set of coordinates for the C_α atoms for each domain — i.e., the spacial coordinates for the carbon atoms linked to the side chains (c.f., Figure 3).



(a) α-helix, most-closely packed arrangement of residues, defined by three parameters: pitch, rise, and turn.

(b) β-sheets form if the backbone is loosely packed, almost fully extended; they can be parallel (left), antiparallel (right), or a mixture.

Figure 8: Secondary structures formed along the backbone define a protein's shape. Dotted lines represent hydrogen bonds; Ⓡ represents a side chain.

To remove a dependency on rigid body motion, structures are often aligned with respect to pairwise distances, d_{ij}, which is a measure between the i^{th} and j^{th} C_α atoms. Let d'_{ij} and d''_{kr} be the intra-distance measures for the two domains, and consider the binary variable

$$x_{ik} = \begin{cases} 1 & \text{if the } i^{th}\ C_\alpha \text{ atom of the first domain is paired with} \\ & \text{the } k^{th}\ C_\alpha \text{ atom of the second domain;} \\ 0 & \text{otherwise.} \end{cases}$$

An optimal pairing between the two domains can be calculated by solving a quadratic integer program:

$$\max \sum_{i,k,j,r} x_{ik} x_{jr} d'_{ij} d''_{kr} : \ \sum_k x_{ik} \leq 1, \ \sum_i x_{ik} \leq 1, \ x_{ik} = 0, \ (i,k) \in \mathcal{S},$$

where $(i, k) \in \mathcal{S}$ if the i^{th} and k^{th} C_α atoms are in different types of secondary structures.

Besides the choice of metric, a variation is to allow pairings between C_α atoms whose secondary structures are different. This is accommodated by removing the restriction that $x_{ik} = 0$ for $(i, k) \in \mathcal{S}$ and adding penalty terms in the objective: $-\sum_{(i,k)\in\mathcal{S}} p_{ik} x_{ik}$. The problem as stated includes the possibility of a non-sequential alignment, i.e., one in which the C_α atoms can be paired independent of the amino acid sequence. A combinatorial optimization model of alignments that requires the same ordering of the amino acid residues is called "contact map optimization" (Burkowski, 2009; Glodzik and Skolnick, 1994; Goldman et al., 1999).

Integer Programming: An integer program (IP) is an optimization problem in which some or all of the variables are restricted to be integer valued. For combinatorial optimization, the integer values are simply $\{0, 1\}$.

Pathway Analysis — Consider the FBA model (3) with added binary variables associated with each process with finite bounds (given or derived), $L_j \le v_j \le U_j$:

$$y_j = \begin{cases} 1 & \text{if } v_j \neq 0; \\ 0 & \text{otherwise.} \end{cases}$$

Replacing the bound constraints with $L_j y_j \le v_j \le U_j y_j$ forces $v_j = 0$ if $y_j = 0$. This corresponds to excluding reaction j, which is called a "knock-out." Drug side-effects are caused by unintended knock-outs, which, if cannot be avoided, can at least be identified and minimized. In drug design, one may want to block all pathways to some final output. If P is a pathway leading to the targeted output, then adding the constraint

$$\sum_{j\in P} y_j \le |P| - 1$$

removes the pathway, where $j \in P$ if pathway P contains reaction j.

A cut set can be computed with successive pathway-generation for a specified output and adding its pathway-elimination constraint. For the example in Figure 2, pathways to produce D can be generated by fixing $v_7 = 1$ (and not have y_7). The first basic optimal solution uses reactions R_1, R_3, R_4, E_1, E_3. This leads to the addition of the constraint:

$$y_1 + y_3 + y_4 + y_5 \le 3.$$

The next pathway generated is R_3, E_1, and $y_3 = 0$ satisfies both pathway constraints. After eliminating R_3, the solution is R_1, R_4, E_1, E_3.

Other logical constraints include process conflict, $y_j + y_{j'} \le 1$ (i.e., inclusion of j requires exclusion of j'), and process dependence, $y_j \ge y_{j'}$ (i.e., exclusion of j requires exclusion of j'), for $j \neq j'$.

Rotamer assignment — Part of the protein folding problem is knowing the side-chain conformations — that is, knowing the torsion angles of the bonds (c.f., Figure 3). The rotation about a bond is called a "rotamer," and there are libraries that give configuration likelihoods, for each amino acid (from which energy values can be derived). The Rotamer Assignment (RoA) Problem is to find an assignment of rotamers to sites that minimizes the total energy of the molecule. For the protein folding problem, the amino acid at each site is known. There are about 10 to 50 rotamers per amino acid, depending on what else is known (such as knowing that the amino acid is located in a helix), so there are about 10^n to 50^n rotamer assignments for a protein of length n.

Let r be in the set of rotamers that can be assigned to site i, denoted by $\mathcal{R}_i$, and let

$$x_{ir} = \begin{cases} 1 & \text{if rotamer } r \text{ is assigned to site } i; \\ 0 & \text{otherwise.} \end{cases}$$

Then, the Quadratic Binary Program (QBP) for the RoA problem is the quadratic semi-assignment problem:

$$\min \sum_i \sum_{r \in \mathcal{R}_i} \left(\mathcal{E}_{ir} x_{ir} + \sum_{j>i} \sum_{t \in \mathcal{R}_j} E_{irjt} x_{ir} x_{jt} \right) :$$
$$\textstyle\sum_{r \in \mathcal{R}_i} x_{ir} = 1 \; \forall \, i, \; x \in \{0,1\}.$$

The objective function includes two types of energy: (1) within a site, $\mathcal{E}_{ir}$, and (2) between rotamers of two different sites, E_{irjt} for $i \neq j$. The summation condition $j > i$ avoids double counting, where $E_{irjt} = E_{jtir}$.

Besides its role in determining a protein's structure, the RoA Problem is useful in drug design. Specifically, the RoA Problem can be used to determine a minimum-energy docking site for a ligand, which is a small molecule such as a hormone or neurotransmitter that binds to a protein and modifies its function. The ligand-protein docking problem is characterized by only a few sites, and if the protein is known, the dimensions are small enough that the RoA Problem can be solved exactly. However, if the protein is to be engineered, then there can be about 500 rotamers per site (20 acids @ 25 rotamers each), in which case solutions are computed with metaheuristics or approximation algorithms. There are other bioengineering problems associated with the RoA Problem, such as determining protein-protein interactions. While the mathematical structure is the same, the applications have different energy data, which can affect algorithm performance (Forrester and Greenberg, 2008).

See (Clote and Backofen, 2000; Jones and Pevzner, 2004; Lancia, 2006) for more.

Dynamic Programming: This is a computational approach to sequential decision-making. Two fundamental biological sequences are taken from the alphabet of nucleic acids, {a,c,g,t}, and from the alphabet of amino acids, {A,R,N,D,C,Q,E,G,H,I,L, K,M,F,P,S,T,W,Y,V}. The former is a segment of DNA (or RNA if u replaces t — i.e., uracil instead of thymine); the latter is a protein segment.

Sequence Alignment — Two sequences can be optimally aligned by dynamic programming, where "optimal" is one that maximizes an objective that has two parts:

1. a *scoring function*, given in the form of an $m \times m$ matrix S, where m is the size of the alphabet. The value of S_{ij} measures a propensity for the i^{th} alphabet-character in one sequence to align with the j^{th} alphabet-character in some position of the other sequence.

   Example: Let $s = \texttt{agt}$ and $t = \texttt{gtac}$. If the first character of s is aligned with the first character of t, then the score is $S_{\texttt{ag}}$, which is the propensity for a to be aligned with g.

2. a *gap penalty function*, expressed in two parts: a "fixed cost" of beginning a gap, denoted G_{open}, and a cost to "extend" the gap, denoted G_{ext}.

   Example: Let $s = \texttt{agt}$ and $t = \texttt{gtac}$. One alignment is $\begin{array}{c}\texttt{agt-}\\ \texttt{gtac}\end{array}$, which puts a gap at the end of the first sequence.

A gap is called an "indel" because it can be either an insertion into one sequence or a deletion from the other sequence: $\underset{\downarrow}{\text{insert}}\ \boxed{\begin{array}{c}\texttt{-}\\ \texttt{a}\end{array}}\ \underset{\uparrow}{\text{delete}}$ If one sequence evolved directly from the other, the evolutionary operation is determined by their time-order. If they have a common ancestor, they evolved along different paths, resulting in the indel when comparing them. The evolutionary biology explains why sequences can be more similar than a simple alignment (without gaps) may suggest.

Figure 9 shows three different alignments for the two nucleic acid sequences, agt and gtac. Scores are shown for the following scoring matrix and do not account for gapping:

$$S = \begin{array}{cccc c} \texttt{a} & \texttt{c} & \texttt{g} & \texttt{t} & \\ \left[\begin{array}{c} 6 \\ 1 \\ 2 \\ 1 \end{array}\right. & \begin{array}{c} 1 \\ 6 \\ 1 \\ 2 \end{array} & \begin{array}{c} 2 \\ 1 \\ 6 \\ 1 \end{array} & \left.\begin{array}{c} 1 \\ 2 \\ 1 \\ 6 \end{array}\right] & \begin{array}{c} \texttt{a} \\ \texttt{c} \\ \texttt{g} \\ \texttt{t} \end{array} \end{array}$$

```
agt--     -a-gt     agt-
 ||        | |      |||
-gtac     gtac-     gtac
```

Score = 12 Score = 2 Score = 4

Figure 9: Three alignments for two sequences.

If the objective is a linear affine function of gap lengths, the total objective function for the 2-sequence alignment problem is:

$$\sum_{i,j} S_{s_i t_j} - G_{\text{open}}(N_s + N_t) - G_{\text{ext}}(M_s + M_t),$$

where the sum is over aligned characters, s_i from sequence s with t_j from sequence t. The number of gaps opened is N_s in sequence s and N_t in sequence t; the number of gap characters (-) is M_s in sequence s and M_t in sequence t. In the example of Figure 9, if G_{open}=2 and G_{ext}=1, the gap penalties are 7, 9, and 3, respectively.

The alphabet is extended to include the gap character, with S extended to include gap extension, as $S_{a\texttt{-}} = S_{\texttt{-}a} = G_{\text{ext}}$ for all a in the alphabet. (So, G_{ext} includes the penalty for the first alignment with -.) Let s^i denote the subsequence $(s_1, \ldots, s_i)$, with $s^0 = \emptyset$. Here is the DP recursion for G_{open}=0:

$$F(s^i, t^j) = \max \begin{cases} F(s^{i-1}, t^{j-1}) + S_{s_i t_j} & \text{match} \\ F(s^{i-1}, t^j) + S_{s_i \texttt{-}} & \text{insert - into } t \\ F(s^i, t^{j-1}) + S_{\texttt{-} t_j} & \text{insert - into } s. \end{cases} \tag{4}$$

The initial conditions are:

$$\begin{aligned} F(\emptyset, \emptyset) &= 0 \\ F(s^i, \emptyset) &= F(s^{i-1}, \emptyset) + S_{s_i \texttt{-}}, \quad i = 1, \ldots, |s| \\ F(\emptyset, t^j) &= F(\emptyset, t^{j-1}) + S_{\texttt{-} t_j}, \quad j = 1, \ldots, |t|. \end{aligned}$$

The DP recursion (4) is for "global alignment," and it has been extended to allow $G_{\text{open}} > 0$ and to not penalize leading or trailing gaps (allowing a short sequence to be aligned with a large one meaningfully). Local alignment is finding maximal substrings (contiguous subsequences) with an optimal global alignment having maximum score (Gusfield, 1997; Waterman, 1995).

Sequences from many species can be compared simultaneously in a Multiple Sequence Alignment (MSA). One way to evaluate an MSA is by summing pairwise scores. Figure 10 shows an example. The sum-of-pairs score, based on the scoring matrix S, is shown for each column. For example, column 1 has $3S_{\texttt{aa}} + 3S_{\texttt{ac}} = 3$. The sum of pairwise scores for column 2 is zero because gap scores are not shown by columns; they are penalized for each sequence (rows of alignment) with G_{open}=2 and G_{ext}=1. The total objective value is $152 - 37 = 115$.

														Gap penalty
	a	-	g	a	g	t	-	a	c	t	-	-	-	11
	a	a	g	t	a	t	-	-	a	t	-	-	-	9
	a	-	-	t	a	t	a	a	-	-	-	-	t	10
	c	-	g	t	a	-	-	a	c	t	c	c	t	7
score:	21	0	18	21	24	18	0	18	8	18	0	0	6	37

Total = 152

Figure 10: A multiple alignment of four sequences.

MSA is a computational challenge to exact DP due to the combinatorial explosion of the state space, but one could use approximate DP or formulate MSA as an IP.

Phylogenetic Tree Construction — Phylogeny is the evolutionary history of some biological entity. A phylogenetic tree (PT) is a graphical presentation of a phylogeny. A leaf represents an Operational Taxonomic Unit (OTU), which can be various levels — e.g., species, genes, pathways, enzymes, microbial communities, bacterial strains. Each edge, or branch, is a relation between pairs of OTUs. Each internal node is constructed so that the resulting PT is consistent with the OTU data, and the root represents a common ancestor of the OTUs.

Example. Consider five OTUs and an MSA of DNA sites with six base-pairs:

	site					
OTU	1	2	3	4	5	6
A	c	a	g	a	c	a
B	c	a	g	g	t	a
C	c	g	g	g	t	a
D	t	g	c	g	t	a
E	t	g	c	a	c	t

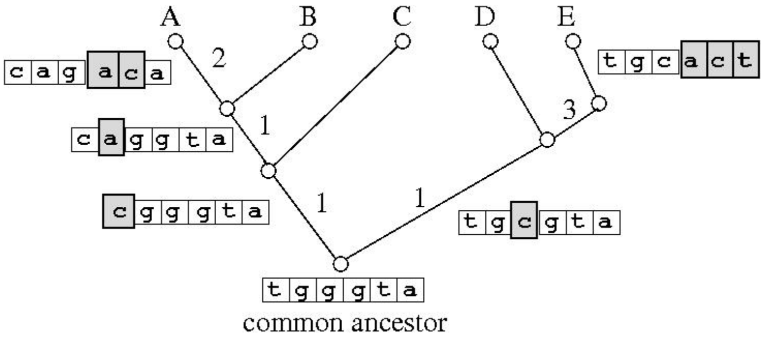


Figure 11: The example maximum-parsimony PT has eight mutations, shown on the branches. (All other PTs have more than 8.)

If the number of mutations is the distance between two sequences, then the distance between OTUs is the length of the unique path between them in the PT. The example has the distance matrix:

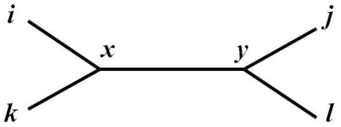
$$D = \begin{array}{cc} \begin{array}{ccccc} A & B & C & D & E \end{array} & \\ \begin{bmatrix} 0 & & & & \\ 2 & 0 & & & \\ 3 & 1 & 0 & & \\ 5 & 3 & 2 & 0 & \\ 8 & 6 & 5 & 3 & 0 \end{bmatrix} & \begin{array}{c} A \\ B \\ C \\ D \\ E \end{array} \end{array}$$

This is not the same as the MSA distance. For example, $D(A,E) = 8$ in the PT but is only 4 in the MSA.

Regardless of how the distance matrix is derived (MSA or not), there may not exist a PT that satisfies specified distances. For that to be true it is necessary and sufficient that the metric be "additive" — i.e., for any four leaves, there exist labels i, j, k, ℓ such that

$$D(i,j) + D(k,\ell) = D(i,\ell) + D(j,k) \geq D(i,k) + D(j,\ell).$$

The reason for this is that there must be some splitting i, k from j, ℓ with an internal branch:



Additivity does not usually hold, so the problem is to construct a PT whose associated leaf-distance matrix, D, minimizes some function of nearness to the given D^0, such as

$||D - D^0||$. This problem is NP-hard. Heuristics include sequential clustering: Unweighted/Weighted Pair Group Method with Arithmetic Mean (UPGMA/WPGMA) and neighbor-joining algorithms.

There may be multiple PTs, which generally come from different data — e.g., one from an MSA of a DNA segment, another from the maximum likelihood of some property. If a series of edge-contractions is applied to a PT, the resulting PT is called a “refinement” and the original is called a “refiner.” Two trees are compatible if they have a common refiner. One problem is to determine whether two PTs are compatible, and if so, what is their common refiner? If incompatible, how is a PT constructed that has some agreement with the given PTs?

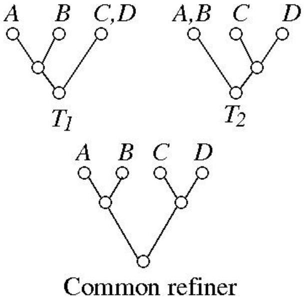


Figure 12: PTs T_1, T_2 are compatible.

A Matrix Representation with Parsimony (MRP) of a PT with k internal nodes is a binary matrix defined as:

$$M_{ij} = \begin{cases} 1 & \text{if internal node } j \text{ is in the (unique) path from the root to OTU } i; \\ 0 & \text{otherwise.} \end{cases}$$

Conversely, given a binary matrix, if it has an associated PT, it is called a “perfect phylogeny.”

Given two PTs for the same OTUs with MRPs, M^1, M^2, their column-union is $[M^1 \; M^2]$.

Theorem. *Two PTs are compatible if, and only if, their MRP column-union represents a perfect phylogeny.*

The trees in Figure 12 have the MRP column-union:

$$M = \begin{array}{c|cl} M^1 & M^2 & \\ 1 & 0 & A \\ 1 & 0 & B \\ 0 & 1 & C \\ 0 & 1 & D \end{array}$$

This is the MRP of the common refiner in Figure 12 and represents a perfect phylogeny.

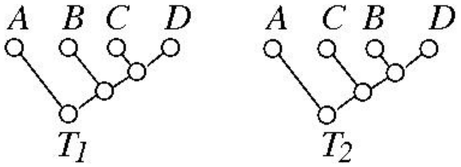


Figure 13: PTs T_1, T_2 are incompatible.

The MRP column-union of the PTs in Figure 13 is:

$$M = \begin{array}{c}\begin{array}{cc} M^1 & M^2 \end{array}\\ \left[\begin{array}{cc|cc} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{array}\right]\end{array}\begin{array}{c} \\ A \\ B \\ C \\ D \end{array}$$

M does not correspond to any PT. (After drawing A, C, D with four internal nodes as the path to D, OTU B cannot be drawn with the path 0-1-3-4 without introducing the cycle, 1-2-3-1.)

Suppose the trees are incompatible. A Maximum Agreement Subtree (MAST) is a refined subtree with the greatest number of leaves.

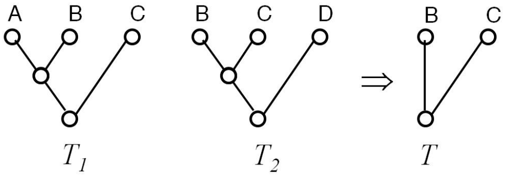


Figure 14: A Maximum Agreement Subtree with 2 of the 4 OTUs.

The DP recursion for two subtrees (Steel and Warnow, 1993) is nontrivial. The state is a pair of subtrees with specified roots, (T_1^r, T_2^s). Each tree has an inclusion-ordered sequence of such subtrees, which is computed during the recursion. The decision space to compute $MAST(T_1^r, T_2^s)$, given $MAST(T_1^{r'}, T_2^{s'})$ for $(T_1^{r'}, T_2^{s'}) \prec (T_1^r, T_2^s)$, requires the computation of a maximum weighted-matching on the complete r-s bipartite graph, weighted with $\{MAST(r', s')\}$.

Whereas MAST uses an intersection of PT information, a supertree uses their union. Construction methods vary, and some of the criteria address common order preservation. An agreement supertree, T, is a minimal tree such that each T_i is a refined subtree of T.

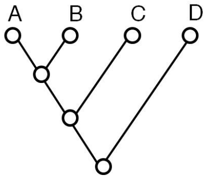


Figure 15: An Agreement Supertree of the trees in Figure 14.

Markov Chains and Processes: A stochastic process has the Markov property if the transition from one state to the next depends on only the current state. Classical models include the evolution of some biological state over time (Allen, 2003). Molecular applications of Markov models also consider ordered sequences of nucleotides (viz., DNA and RNA) and amino acids (viz., proteins).

CpG island recognition — In the human genome the appearance of the dinucleotide CG is rare because it causes the cytosine (C) to be chemically modified by methylation, which causes it to mutate into thymine (T). Methylation is suppressed around the promoters, or start regions, of many genes, and there are more CG dinucleotides than elsewhere. Such regions are called "CpG islands," and they are typically a few hundred bases long. (CpG is used instead of CG to avoid confusion with a C-G base pair; the p is silent.) The recognition problem is: Given a short segment of a genomic sequence, decide if it is part of a CpG island.

Two Markov chains are defined: P^+ is the state-transition matrix within a CpG island; P^- is the state-transition matrix outside a CpG island. Each is applied to the given sequence and the log-odds ratio determines which is more likely.

Example. Consider a first-order Markov chain model with transition matrices determined by the frequencies in a database having more than 60,000 human DNA sequences:

$$P^+ = \begin{array}{c} \begin{array}{cccc} \mathtt{A} & \mathtt{C} & \mathtt{G} & \mathtt{T} \end{array} \\ \begin{bmatrix} 0.18 & 0.27 & 0.43 & 0.12 \\ 0.17 & 0.37 & 0.27 & 0.19 \\ 0.16 & 0.34 & 0.38 & 0.12 \\ 0.08 & 0.36 & 0.38 & 0.18 \end{bmatrix} \end{array} \qquad P^- = \begin{array}{c} \begin{array}{cccc} \mathtt{A} & \mathtt{C} & \mathtt{G} & \mathtt{T} \end{array} \\ \begin{bmatrix} 0.30 & 0.20 & 0.29 & 0.21 \\ 0.32 & 0.30 & 0.08 & 0.30 \\ 0.25 & 0.25 & 0.30 & 0.20 \\ 0.18 & 0.24 & 0.29 & 0.29 \end{bmatrix} \end{array}$$

Given the sequence AACTTCG, its total log-odds ratio is

$$\sum_{i=1}^{6} \log_2 \left(P^+_{s_i s_{i+1}} / P^-_{s_i s_{i+1}} \right) = -0.737 + 0.433 - 0.659 - 0.688 + 0.585 + 1.755 = 0.6888.$$

The conclusion is that the DNA segment is in a CpG island.

There is enough data to support the use of the more-accurate 5th-order Markov chain, whose 6-tuples correspond to two coding regions. At least 4^5 6-tuples are required in the database to estimate the conditional probabilities, $\Pr(x_6 \mid x_1x_2x_3x_4x_5)$, which directly yield the state-transition probabilities:

$$\Pr(y_1y_2y_3y_4y_5 \mid x_1x_2x_3x_4x_5) = \begin{cases} \Pr(x_6 \mid x_1x_2x_3x_4x_5) & \text{if } y = (x_2x_3x_4x_5x_6); \\ 0 & \text{otherwise.} \end{cases}$$

For the particular example, there are only two state transitions, and the same database gives the transition probabilities:

$$\begin{array}{ll} P^+(\mathtt{C} \mid \mathtt{AACTT}) = 0.4 & P^-(\mathtt{C} \mid \mathtt{AACTT}) = 0.2 \\ P^+(\mathtt{G} \mid \mathtt{ACTTC}) = 0.1 & P^-(\mathtt{G} \mid \mathtt{ACTTC}) = 0.3 \end{array}$$

In this case the more accurate 5th-order chain yields the log-odds ratio $\log_2 0.4/0.2 + \log_2 0.1/0.3 = -0.585$, and the conclusion is that the DNA segment is ***not*** in a `CpG` island.

A host of related problems use the same Markov model. For example, transcription splices the DNA into coding regions, called "exons," removing the remainder, called "introns" (misnamed "junk DNA"). A structure recognition problem is to identify exons vs. introns.

Many of the structure recognition, comparison, and prediction problems have hidden states, but emissions are observed according to a known probability. These are "Hidden Markov Models" (HMMs) and are central in modern biology (Durbin et al., 1998).

Queueing Theory: A queue in a system is any set of objects awaiting service, and service is some process(es) involving the object.

T-cell signaling — A T-cell is a type of white blood cell distinguished by having a *receptor* — an ability to bind to other molecules. The receptor interacts with intracellular pathway components, starting a cascade of protein interactions called "signal transduction." A way to view this process is that a T-cell receptor (TCR) enters a queue upon activation and goes through a series of processes, such as phosphorylation (Wedagedera and Burroughs, 2006). Service completion is defined by the deactivation of the TCR, returning it to the inactive pool; however, it is possible that the T-cell's service is aborted before it completes service. Of interest is the probability of activation — i.e., in service for some threshold of time. If it completes service and detects infection, the T-cell signals cell death (called "apoptosis," pronounced ăp'ō-tō'sĭs; the 'p' is silent).

Other queueing models apply to genetic networks, allowing signals that affect the population to enter and leave the system (Arazi et al., 2004; Jamalyaria et al., 2005).

This applies queueing to a broad range of *self-assembly* systems — i.e., form an arrangement without external guidance.

Simulation: Dynamical state evolution is fundamental in both classical mathematical biology and modern systems biology. Evolution and biochemical pathways are prime examples; the underlying state-transition structure and the sheer size are sufficient to need simulation.

The kinetic laws of a biosystem depend upon the objects, particularly their scale (viz., molecules vs. cells). The deterministic rate equations have the form:

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = f_i(x; k) \qquad\qquad \text{for } i = 1, \ldots, m,$$

where x is the system state (e.g., concentrations of m metabolites) and k is a vector of parameters, called *rate constants.*

Sources of randomness can be *intrinsic* — e.g., errors in parameter estimation, or *extrinsic* — e.g., protein production in random pulses (Meng et al., 2004). To deal with reaction uncertainty, Gillespie (2008, 1977) introduced the probability equation:

$$\Pr(x; t + \mathrm{d}t) = \textstyle\sum_r a_r(x - v_r)\,\mathrm{d}t + \Pr(x; t)\,(1 - \sum_r a_r(x)\,\mathrm{d}t)\,,$$

where $a_r(x)\,\mathrm{d}t$ is the probability that reaction r occurs in the time interval $(t, t + \mathrm{d}t)$, changing the state from x to $x + v_r$. The first summation represents being one reaction removed from the state x; the last term represents having no reaction during the interval.

Auto-regulatory network — Puchalka and Kierzek (2004) consider a metabolic network with regulatory processes and random fluctuations in gene expression. Using Gillespie's equation, given the state x at time t, the probability that the next reaction, r, occurs during $(t + \tau, t + \tau + dt)$ is given by:

$$\Pr(\tau, r \mid x, t) = a_r(x)\, e^{-\sum_j a_j(x)\tau}.$$

The simulation is run by generating (τ, r) using this joint density function. The simulation also allows for pulse production — a receptor site may be on or off to regulate gene expression (restricting the choice of r).

Other models use rare-event simulation, such as for tumor development (Abbott, 2002). Simulation is used in systems biology to understand how non-dominant pathways affect assembly kinetics (Zhang and Schwartz, 2006).

Game Theory: The central idea of game theory is that each player has its own objective to optimize. Historically, evolutionary biologists used game theory to model natural selection (Maynard Smith, 1982; Perc and Szolnoki, 2010). In OR, game theory is used to model competition for economic resources, and this extends to modeling species-invasion into an existing ecosystem. The same game model applies to propagation of tumor cells that can mutate in minutes to create a cancer population that overwhelms normal cells (Tomlinson, 1997). New applications are at the molecular scale, such as the following example.

Protein binding — There are two sets of players: protein classes (including drugs) and DNA binding sites. Their joint strategies result in allocation of proteins to sites. Sites seek to maximize their occupancy; proteins seek to minimize excess binding. Sites compete for nearby proteins; proteins choose target sites to which they transport. (Mechanisms to achieve these choices are not well understood.) The affinity for protein i to bind to site j is denoted by the constant K_{ij}, but this applies only if the protein is in the proximity of the site.

Let $i = 1, \ldots, N_p$ index proteins and $j = 1, \ldots, N_s$ index sites, and consider the parameters:

$$\begin{aligned} \nu_i &= \text{nuclear concentration,} \\ E_{ij} &= \text{transport affinity,} \\ K_{ij} &= \text{binding affinity.} \end{aligned}$$

A protein's decision variable is its fractional transported amounts, $p^i = (p_0^i, \ldots, p_{N_s}^i)$, where $p_0^i = 1 - \sum_{j=1}^{N_s} p_j^i$ is the portion of protein i not allocated to a site. A site's decision variable is its choice of binding frequency, $s^j = (s_0^j, \ldots, s_{N_p}^j)$, where $s_0^j = 1 - \sum_{i=1}^{N_p} s_j^i$ is the portion of time that site j is unoccupied. There are resource constraints on joint strategies, notably $s_i^j \le p_j^i \nu_i$ for $i > 0$ — i.e., binding cannot exceed allocated concentration.

A solution is a joint strategy $(\overline{p}, \overline{s})$ that satisfies the optimality criteria:

$$\overline{p}^i \in \underset{p^i \in P(\overline{s})}{\operatorname{argmax}} \{ f_p^i(p^i, \overline{s}) \} \qquad\qquad \overline{s}^j \in \underset{s^j \in S(\overline{p})}{\operatorname{argmin}} \{ f_s^j(\overline{p}, s^j) \},$$

where f_p, f_s denote objective functions for each protein and site, and $P \subseteq \mathbb{R}_+^{N_s+1}$, $S \subseteq \mathbb{R}_+^{N_p+1}$ denote feasible regions, each dependent on the other decisions. An example of objective functions are maximizing total binding affinity and minimizing the amount of protein not assigned:

$$\begin{aligned} f_p^i(p^i, s) &= \sum_{j=1}^{N_s} E_{ij} p_j^i (1 - s_0^j) \\ f_s^j(s^j, p) &= s_0^j \sum_{i=1}^{N_p} K_{ij} (p_j^i \nu_i - s_i^j). \end{aligned}$$

With mild modifications, a solution exists and there is a simple algorithm to find it (Pérez-Breva et al., 2006).

This game model is a simplification of a broader biology, where sites can coordinate, not just compete, and proteins can form complexes to bind to the same site. There are also promoters that bind to a protein in order to send it to another site. Although current thinking is that proteins roam randomly until they bump into an unoccupied site for which they have affinity, the game model attributes a purposeful behavior to proteins, suggesting that they choose to transport to some site. While this rational behavior is not due to intelligence, it could be due to an environmental context that is not yet understood and whose net effect makes proteins behave ***as if*** they are rational players.

References

R. Abbott. CancerSim: A computer-based simulation of Hanahan and Weinberg's Hallmarks of Cancer. Masters thesis, The University of New Mexico, Albuquerque, NM, 2002.

L. J. S. Allen. *An Introduction to Stochastic Processes with Applications to Biology.* Pearson Education, Upper Saddle River, NJ, 2003.

A. Arazi, E. Ben-Jacob, and U. Yechiali. Bridging genetic networks and queueing theory. *Physica A: Statistical Mechanics and its Applications*, 332:585–616, 2004.

F. Burkowski. *Structural Bioinformatics: An Algorithmic Approach.* Mathematical and Computational Biology. Chapman & Hall/CRC, Boca Raton, FL, 2009.

P. Clote and R. Backofen. *Computational Molecular Biology.* John Wiley & Sons, New York, NY, 2000.

R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids.* Cambridge University Press, Cambridge, UK, 1998.

C. A. Floudas and P. M. Pardalos, editors. *Optimization in Computational Chemistry and Molecular Biology: Local and Global Approaches.* Kluwer Academic Publishers, 2000.

R. J. Forrester and H. J. Greenberg. Quadratic binary programming models in computational biology. *Algorithmic Operations Research*, 3(2):110–129, 2008.

D. T. Gillespie. Simulation methods in systems biology. In M. Bernardo, P. Degano, and C. Zavattaro, editors, *Formal Methods for Computational Systems Biology*, LNCS 5016, pages 125–167. Springer, 2008.

D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

A. Glodzik and J. Skolnick. Flexible algorithm for direct multiple alignment of protein structures and sequences. *Bioinformatics*, 10(6):587–596, 1994.

D. Goldman, S. Istrail, and C.H. Papadimitriou. Algorithmic aspects of protein structure similarity. In *40th Annual Symposium on Foundations Of Computer Science (FOCS)*, pages 512–521. IEEE Computer Society Press, 1999.

D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, Cambridge, UK, 1997.

F. Jamalyaria, R. Rohlfs, and R. Schwartz. Queue-based method for efficient simulation of biological self-assembly systems. *Journal of Computational Physics*, 204(1):100–120, 2005.

N. C. Jones and P. A. Pevzner. *An Introduction to Bioinformatics Algorithms*. MIT Press, Cambridge, MA, 2004.

G. Lancia. Applications to computational molecular biology. In G. Appa, P. Williams, P. Leonidas, and H. Paul, editors, *Handbook on Modeling for Discrete Optimization*, volume 88 of *International Series in Operations Research and Management Science*, pages 270–304. Springer, 2006.

J. Maynard Smith. *The Theory of Games and the Evolution of Animal Conflicts*. Cambridge University Press, Cambridge, UK, 1982.

T. C. Meng, S. Somani, and P. Dhar. Modeling and simulation of biological systems with stochasticity. *In Silico Biology*, 4(0024), 2004.

B. Ø. Palsson. *Systems Biology: Properties of Reconstructed Networks*. Cambridge University Press, New York, NY, 2006.

M. Perc and A. Szolnoki. Coevolutionary games — A mini review. *BioSystems*, 99(2): 109–125, 2010.

L. Pérez-Breva, L. E. Ortiz, C-H. Yeang, and T. Jaakkola. Game theoretic algorithms for protein-DNA binding. In *Proceedings of the 12th Annual Conference on Neural Information Processing (NIPS)*, Vancouver, Canada, 2006.

J. Puchalka and A. M. Kierzek. Bridging the gap between stochastic and deterministic regimes in the kinetic simulations of the biochemical reaction networks. *Biophysical Journal*, 86(3):1357–1372, 2004.

M. Steel and T. Warnow. Kaikoura tree theorems: Computing the maximum agreement subtree. *Information Processing Letters*, 48(3):77–82, 1993.

I. P. M. Tomlinson. Game-theory models of interactions between tumour cells. *European Journal of Cancer*, 33(9):1495–1500, 1997.

M.S. Waterman. *Introduction to Computational Biology: Maps, Sequences, and Genomes (Interdisciplinary Statistics)*. Chapman & Hall/CRC, Boca Raton, FL, 1995.

J. R. Wedagedera and N. J. Burroughs. T-cell activation: A queuing theory analysis at low agonist density. *Biophysical Journal*, 91:1604–1618, 2006.

T. Zhang and R. Schwartz. Simulation study of the contribution of oligomer/oligomer binding to capsid assembly kinetics. *Biophysical Journal*, 90:57–64, 2006.