8 Utilizing markov chains to model ion channel sequence variation and kinetics

Anthony Fodor

Contents

8.1 INTRODUCTION: SEQUENCE VARIATION DRIVES DIFFERENCES IN CHANNEL BIOPHYSICS, BUT LINKING THIS VARIATION TO HUMAN PHENOTYPE VARIATION REMAINS A CHALLENGE

In their Nobel Prize–winning formulation of the squid giant axon, Hudgkin and Hulxey [1] showed how quantitative models could be applied to describe how ion currents shaped electrical signaling. In the Hodgkin and Huxley models, parameters for activation and inactivation were fit from experimentally observed electrophysiological data. With the advent of molecular biology in the late 1980s, ion channels were cloned, and full-length sequences become known for many ion channels [2–5]. It was a natural question to ask how sequence variation in ion channels was linked to variation in parameters that described their electrical properties. With the introduction of site-directed mutagenesis and the ability to express ion channels in exogenous systems such as *Xenopus* oocytes [6], the effect of changes in sequence could be directly measured. This approach of introducing variants into a channel sequence and observing the results for channel function has had many successes in a wide variety of channels and has produced a detailed understanding of how sequence variation can constrain channel function. For example, in *Shaker* potassium channels, site-directed mutagenesis demonstrated that inactivation was tied to the amino-terminus region [7]. In the CFTR channel, distinct mutations associated with loss of channel function were determined to be causative of most types of cystic fibrosis [8].

While specific mutations in crucial regions can be linked to dramatic changes in channel function and disease, mutations throughout an entire ion channel's sequence can often affect ion-channel energetics in subtle ways. Studies that have utilized chimeras between related channels have found a modular design to channels in which different regions of the channel cooperate to influence channel energetics. For example, in cyclic-nucleotide-gated (CNG) channels, it has been demonstrated that a channel derived from olfactory epithelium is much more prone to favor an open state when compared with a homologous channel from rod cells. Chimeric channels that were intermediate in sequence between rod and olfactory channels had intermediate energetics with no one region of the channel sequence completely controlling the energetics of channel opening [9,10]. These results suggest that mutations that tune channel function can be spread throughout the channel sequence and that sequence variants found in sequence databases may cause subtle changes to channel energetics that may nonetheless be important in determining channel phenotype and function.

At the turn of the century, the human genome project was completed, initiating the era of genomics. While the first human genome project took decades of work with costs running into the billions of dollars, recent advances in next-generation sequencing have brought us to the edge of the *thousand dollar genome*. We can look forward to a near future in which the genomes of every subject in a clinical trial are known. A central challenge for all families of genes in the postgenomic era is to link changes in genome sequence to functional consequences in health and disease. Since mutations throughout the channel sequence can tune channel function in ways that presumably have profound consequences for host phenotype, it would be natural to think that whole-genome association studies, that link genotypic changes to phenotypes of interest, would yield a rich repository of important changes to ion channels that could explain diseases, such as heart disease and psychiatric disorders, that we know must involve ion channels. The expectation that whole-genome association studies would substantially explain complex diseases, however, has in general not been realized. In study after study, it has been found that variations described by whole-genome association studies have only been able to explain a small percentage of human phenotypic variation [11]. For example, a recent meta-analysis looked at genome-wide association studies (GWAS) that examined five neurological disorders: autism spectrum disorder, attention deficit-hyperactivity disorder, bipolar disorder, major depressive disorder, and schizophrenia [12]. Intriguingly, single nucleotide polymorphisms (SNPs) within two L-type voltage-gated calcium channels were significantly associated with several of these disorders. Not only does a study of this sort suggest these calcium channels as a possible drug target but also suggest that a single genetic mechanism might underlie a series of diseases that are currently categorized in the clinic in distinct ways. This raises the possibility that in the near future genetic approaches may allow for more clear and consistent diagnoses that are possible by only considering phenotype. However, as is the case in nearly all GWAS [13], while the contribution of the calcium channels to these disorders is reproducible across studies and statistically significant, the relative risk that carriers of mutant alleles of these channels have for the psychiatric disorders is modest. Put another way, the presence of these mutant channel alleles explains very little of who in the population does or does not present disease.

For most complex diseases, as in these psychiatric disorders, it is apparent that there is a great deal of human variation that cannot be modeled as an additive sum of independent changes in gene sequence from common alleles. There has been feverish interest in determining the source of the *missing heritability* that is generally not captured by GWAS studies [11,14]. Because most GWAS studies have focused on common alleles, one possible explanation for the low power of GWAS studies is that most human variation in complex diseases is in fact caused by low frequency alleles. As SNP arrays become replaced by ever more affordable full genome resequencing, it will become increasingly possible to measure rare variants. As more genomes are sequenced from more people allowing for more genomic detail for larger sample sizes, we will learn to what extent rare variants contain the missing variability. In addition, as technologies continue to develop that allow for measurement of genome structure that have not necessarily been captured by SNP arrays—such as copy number variation and epigenetic changes—we will learn to what extent genome variation not described by simple sequence explains the missing heritability. Gene–gene interactions [14] and gene–environment interactions [15,16] may also explain the low power of GWAS studies. If the missing heritability is caused by these sorts of complex interactions, then understanding the link between sequence variation and consequences for health in all genes families—including ion channels—will require new experimental paradigms that consider the context in which sequence variation within an individual gene occurs.

While whole-genome associations studies look for variation within a species to try and link sequence variation to function, it is also possible to look across species to track how evolutionary changes across long time spans constrain channel function. This approach is dependent upon finding conserved ion channels in organisms that may be distantly related. In this chapter, we examine fundamental algorithms that allow for detection of ion channels in large sequence databases. We note that some of these algorithms may already be familiar to students of ion channels as the hidden Markov Model (HMM) formalism that describes the behavior of ion channels can easily be modified and applied to sequence alignments. We conclude with a discussion of how sequence variation is likely to be studied in the future within the context of gene–gene and gene–environment interactions.

8.2 THERE ARE HIGHLY EFFICIENT ALGORITHMS FOR FINDING ION CHANNELS IN LARGE DATABASES

As the cost of sequencing continues its exponential drop, the number of distinct proteins that have been identified has greatly increased. A dramatic example of this phenomenon was the Venter Institute's Sorcerer II global ocean survey, which generated 7.7 million sequencing reads from 41 ocean sites [17]. At the time of publication in 2007, this was the largest survey of mixed environmental microbial communities. Remarkably, this single sequencing effort nearly doubled the number of proteins known at the time [18]. With the increasing prevalence of next-generation sequencing, this tremendous increase in sequence diversity of proteins from across the phylogenetic tree has only continued.

Given a query sequence, say an ion channel or an ion-channel fragment, and a large database of sequences to search, the most basic bioinformatics task is to find sequences similar to the query within the database. The execution of such an algorithm allows us to discover new channels in a newly sequenced organism given knowledge of existing channels. These *homology detection algorithms* come in two flavors: searches based on *global alignments* attempt to find the entire query sequence in the database while searches based on *local alignments* look for a significant match of any part of the query sequence, for example, that of the potassium channel selectivity filter. Because sequence reads from Sanger sequencing or next-generation sequencing are often shorter than reads for an entire gene, especially for long genes like ion channels, and because evolution may have only conserved part of a protein across multiple species, algorithms based on local alignment are often more useful than those based on global alignments.

The most commonly used local alignment algorithm is BLAST [19], developed over 25 years ago, but still indispensible for many bioinformatics applications. BLAST is an example of a *pairwise alignment* algorithm, meaning it compares the query sequence to each sequence in the database. Pairwise alignment algorithms often utilize a *substitution matrix*, which defines the similarity of symbols between sequences so that, for example, an alanine in the query sequence will be more likely to be matched to a glycine in the target rather than to a proline residue. Substitution matrices are empirically derived from sets of aligned sequences. Many implementations of BLAST default to the BLOSUM62 matrix [20], which was built on a training set of proteins with a threshold of a 62% identity. For all 20 possible pairs of amino acids, the frequency of each substitution within the training set is observed and converted to a log probability of a match divided by the background frequency of the residue pair. Like any data constructed with a training set, the choice of a substitution matrix captures or summarizes our expectations of what it means for two proteins to be similar. Since default training sets will be built from many different protein families, and not just membrane proteins or ion channels, these choices in principle can affect what is detected when using a membrane or ion-channel query sequence. Substitution matrices derived from membrane proteins may therefore be more sensitive for use in ion-channel homology detection than generic substitution matrices [21]. In practice, however, generic substitution matrices usually offer good enough performance in finding ion channels in databases of newly sequenced organisms.

Given a query sequence, a target sequence, a substitution matrix, and some strategy for how gaps in an alignment are scored, algorithms that exploit *dynamic programming* can generate an alignment between the target and the query that guarantees that no *better* alignment exists, where a *better* alignment would be one in which more similar residues as defined by the substitution matrix are aligned to each other (see [22] for an essential overview). These dynamic programming alignment algorithms (Needleman–Wunsch [23] for global alignments and Smith–Waterman [24] for local alignments) work by first building the best subalignment from the initial residues of the proteins and then recursively expanding on the subalignment. Crucial to this approach is the *assumption of independence*, which assumes that once a subalignment has been scored, no further residues that are later added to the subalignment can change the score of the subalignment. In terms of protein structure, this assumption could be interpreted as saying that no two regions of a protein interact. While this is clearly not a good assumption for ion channels, or any other complex protein, in practice, dynamic programming algorithms tend to give very reasonable alignments, although they will be insensitive to long-range interactions. So, for example, if a pore region of one ion channel is consistently linked to a ligand-gating region but these regions are separated by a long nonconserved region, the information that the two regions should only be found together cannot be used by dynamic programming algorithms to make a search more sensitive since the two subregions of the protein will be scored in an independent matter.

While dynamic programming algorithms guarantee the alignment with the *best* score, subject to the assumption of independence, for a given query and target sequence, they are generally too slow to be of much use in large-scale sequence analysis. The problem is that given a large database and a query sequence, the query sequence must be compared to every protein in the database, one at a time. We say that dynamic programming *scales* in terms of the size of the database, meaning that the larger the database gets, the longer it takes to search the database. Given that modern genomics databases can easily have terabases of sequence, this is far too slow to be practical. What is needed, instead, are algorithms that scale in terms of the size of the query sequence. As more sequences accrue, the databases become bigger every year, but the length of the genes themselves does not grow. It turns out that there are many data structures (including suffix trees and hash maps) that allow an *exact match* search to be performed in time proportional to the query rather than the database. That is, given a database of arbitrary size (subject to the amount of memory available in your system), it is possible to find an exact match from a query string in time that is proportional to the length of the query, not the size of the database. This always involves a *preprocessing* step in which some kind of dictionary is built from the database. This preprocessing step is often very slow, but only has to be done once. Once it is finished, queries against the database can be performed very quickly. This central trick in computer science is how search engines like Google are able to search the entire web and return results back from a query essentially instantly. The web is big (and growing), but the queries we type are small (and do not grow over time). Likewise, the number of sequences in databases is large (and growing), but the lengths of the queries (e.g., the length of ion-channel genes) do not increase over time. By using algorithms that depend on the length of the query and not the length of the database, our algorithms remain usable even as the universe of data continues to rapidly expand.

Algorithms like BLAST, therefore, use *heuristics* (time-saving approximations) that render them less sensitive to the size of the database. These heuristics mean that we give up the guarantee that our hit will be the best possible hit relative to a substitution matrix, but they allow the algorithms to execute in a reasonable amount of time. The most common heuristic is to constrain searches so that they will only succeed if there is some number of exact substring matches between the query and the target. This no longer guarantees the theoretically best hit and will ignore features such as chemical similarity since, for example, a search dependent on exact substrings cannot score a glycine-to-alanine substitution higher than glycine-to-proline. A type of search known as *k-mer* is dependent on the choice of a *word size* (e.g., three residues) and breaks the query sequence into all possible words of that length (in our example, every *word* of three residues within our protein). For each word, the database is searched, which can be done in *constant time* (i.e., in time independent of the size of the database). The top hit is the one that has the most k-mers (in our example 3 mers) in common between the query and the target. K-mer searches are fast but by themselves do not produce an alignment. Moreover, if the query and the protein target do not have many regions of identity that are equal in length to the word size, a k-mer search may miss proteins that are in actuality homologs. BLAST works by combining some of the features of a k-mer search with some of the features of dynamic programming. BLAST starts by selecting a word length (by default 3 for protein searches and 11 for nucleotide searches). Then for target sequences with matching k-mers, BLAST uses a dynamic programming like extension step to try and build an alignment from the k-mer seed. By changing the word length, a user can adjust speed versus sensitivity. A longer word length will be less sensitive, requiring a longer region of exact match, but will also be faster as the extension step will be executed on fewer matches. A shorter word length will be more sensitive but slower as the extension step will be executed on more matches. Many derivatives and alternatives to BLAST have been developed, including the popular algorithm BLAT [25], which can be both faster and more memory efficient than BLAST.

While word size can have a profound effect on the results of a BLAST search, there are a number of other important parameters that can be set by the users. This includes which substitution matrix is used, how gaps are treated within the alignment, and whether searches are to be performed at the DNA or protein level. Because of the degeneracy of the genetic code, searches for distantly related ion channels can often fail at the nucleotide level but will succeed if both query and target sequences are translated into all six frames and the search is performed in protein space. These translated searches, however, can also take longer and can be more sensitive to low-complexity regions of the genome and other artifacts that can produce erroneous results. Using a filter to remove low-complexity regions of the genome can therefore be essential and is an option for many BLAST implementations.

Results of BLAST searches are typically ranked by their significance. BLAST results usually have an alignment score and an e-value, defined as the number of hits with at least that alignment score that one might expect by chance when searching a database of the same size with a query sequence of the same size. By this definition, an e-value >1 represents a hit that is worse than one would expect by chance while an e-score <1 could be considered to be significant. However, these e-values must be interpreted with due caution. The accuracy of the e-values is dependent on a model of the composition of a random sequence. AQ1 The e-values can be thought of representing how likely one is to find a protein of the same length to be the query sequence make up of *random* residues, but the model that was used to define *random* residues will likely have been built on all proteins, not just membrane proteins or ion channels. E-values should therefore be considered only a rough guide to the significance of the results.

In general, the results of BLAST searches should be considered as an experiment dependent on a particular set of parameters and some understanding of how these parameters effect BLAST results can improve the chances of detecting homology across a wide phylogenetic space. There are excellent resources available discussing how these parameters can be tuned or particular searches (see, e.g., [26]). Blast searches can AQ2 be run at many websites (e.g., at NCBI—http://blast.ncbi.nlm.nih.gov/Blast.cgi) or the BLAST software can be downloaded (http://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=BlastDocs&DOC_TYPE=Download) and used to set up a database on almost any computer.

8.3 HIDDEN MARKOV MODELS ARE A FLEXIBLE FRAMEWORK THAT CAN BE USED TO MODEL MANY DIFFERENT PROBLEMS

Because of their speed and flexibility, pairwise alignment algorithms remain a workhorse in bioinformatics. However, it was realized some time ago that there may be information available within multiple sequence alignments that is not capturable by a pairwise approach, in particular, when the sequence similarity between the two sequence is low [19,27]. Rather than comparing a query sequence to every sequence in a database and returning the best hit, homology detection algorithms based on *profiles* build a model based on a *multiple sequence alignment* of sequences from the same family. It has been shown that by exploiting information from these alignments, searches conducted with these profiles can be more sensitive than pairwise searches [19,27].

The homology detection approach based on profiles is perhaps best exemplified by the popular PFAM database [28,29]. Rather than use a global substitution matrix, PFAM works by building a model in which each column within the multiple sequence alignment has its own expected distribution of amino acid residues. Using this model, one can ask how well a query sequence matches the protein family represented by the sequences within the multiple sequence alignment.

In order to build these probabilistic models that describe protein alignments, PFAM utilizes Markov chains. Markov chains, and the algorithms that manipulate them, are remarkably useful tools that can describe an extraordinary diversity of phenomena. Markov chains are utilized in every area of quantitative science, from speech recognition to statistical mechanics. As we will see later, Markov chains provide great flexibility in modeling and we will demonstrate their use in simple examples for both ion-channel sequence variation and ion-channel kinetics.

Markov chains can be used to model any set of data that is ordered. A Markov chain consists of a set of interconnected Markov states. Each observed piece of data is considered to be *emitted* from some state of the Markov chain. The data point at each interval is observed and known; however, which state the Markov model was in when the data was observed is not necessarily known. When the state the model is in corresponding to each emission is unknown, the Markov model is called hidden, hence the term *hidden Markov models* (*HMMs*). Each Markov state has an *emission probability*, which is the probability that data will be observed given that the model is in that state. States in a Markov chain are linked together by *transmission probabilities*, which is the probability that one moves from one Markov state to another. A key feature of Markov chains that we will consider is that they do not have memory; the probability of being in a particular Markov state is dependent only on the current state.

8.4 MARKOV MODEL OF SINGLE ION-CHANNEL KINETICS

As an example of the flexibility of Markov chains, we will consider their application to ion-channel kinetics. We will here only consider a highly simplified model free from *real-world* problems such as baseline drift. The application of Markov models to real ion-channel data is discussed in detail elsewhere [30–32]. In our simplified model, we consider an ion channel that has only two states, closed and open (Figure 8.1 top panel). The rate parameter α defines the rate at which the channel moves from the closed state to the open state while the parameter β defines the rate at which the channel moves from the open state to the closed state. We say that the states of the channel display *independence* and do not have *memory*. In this simple two-state model, when the channel is in the closed state, it has no memory of how long it has been in the closed state or how many times the closed to open transition has occurred.

We can use a Markov chain to model the kinetics of our simple two-state channel (Figure 8.1, bottom panel). The Markov chain also has two states, closed and open. A Markov chain has a concept of an iteration, which in this case represents some short unit of time.* Rather than rate constants α and β,

* A convenient unit of time in a simulation of single-channel data might be the sampling time of the A/D converter used to acquire the single-channel data.

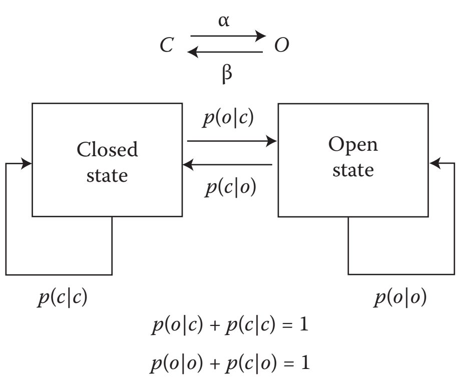


Figure 8.1 A simple two-state model of an ion channel capable of simulating single-channel recordings. Top panel: A chemical model with rate constants α and β. Bottom panel: The equivalent Markov chain with four transition parameters.

a Markov model defines *transition probabilities*. In our simple two-state model, there are four transition probabilities. If we are in the closed state, we can in the next iteration stay in the closed state. This transition probability is $p(C|C)$ which we read as *the probability that we stay in the closed state given that we are in the closed state*. Or alternatively, we can transition from the closed state to the open state, which is defined as $p(O|C)$ or *the probability that we move to the open state, given that we are in the closed state*. Obviously, in our simple model, $p(C|C) + p(O|C) = 1$, since in our two-state model, we must either stay in the closed state or move to the open state. We can likewise define transition probabilities from the open state, $p(C|O)$ and $p(O|O)$, which also must sum to one. This first-order Markov chain, like our chemical model, has no memory. The model only knows its current state, not its history. This feature also establishes independence as the behavior in a given state is independent of the history of how that state was achieved. These assumptions greatly simplify the implementations of algorithms that define and decode Markov chains.

In addition to transition probabilities, Markov chains also define *emission probabilities*. At each iteration, when the model is in a given state, it emits a value from some alphabet or distribution. As we will see, in Markov chains designed to model protein families, the emission symbol is one of the 20 protein residues or, possibly, a gap. For our simple ion-channel model, the emission probability describes the probability of observing a given current. For each state, the emission probabilities over all possible emissions
AQ3 must sum to one. We can choose any distribution of probabilities, but a convenient distribution is a simple Gaussian distribution:

$$\frac{1}{\sqrt{2}}e^{-\frac{1}{2}(\frac{x-u}{})^2}.$$

AQ4 the μ and σ the mean and standard deviation of the current at each state under some conditions for recording and x the current actually observed (see [31] for a discussion of HMMs with more sophisticated noise models). Given a set of emission parameters and transition parameters, it is straightforward to generate a simulated set of emissions. One simply chooses a start state* and samples an emission as defined by the emission probabilities of that state. Then, one either moves to the other state or stays in the same state as defined by the transition probabilities of that state. Then this process is repeated. The results of such a run for our two-state ion-channel model (current as a function of iteration/time) are shown in Figure 8.2. (The code used to generate this figure is available at https://github.com/afodor/IonChannelHMMs.)

We see that despite the great simplicity of our model, we can get surprisingly realistic single-channel data. By adding additional states to the model, we can increase the complexity of our simulated channels behavior. We can, for example, add states that represent long-lived inactivation and bursting (Figure 8.3).

In generating the simulated dataset, we know the state during each iteration, which forms a chain of states called a *path*, and use that information to produce a sequence of ordered emissions. The great strength of HMMs is that we can go in the opposite direction; that is, given just the emissions and a model with a given set of parameters, we can calculate the probability of being in each state for each iteration. This calculation is called *posterior decoding* (see [22]). For our ion-channel models, given the emissions that come from a *hidden* state in an HMM, we can efficiently calculate the probability that the hidden state was open or closed for each iteration. We say that given the data (and a Markov chain that is our model), we can calculate the p(open|data) and p(closed|data) for every emission. In the case of the model parameters shown in Figure 8.2, where there is no overlap between the open and closed states, this posterior decoding is perfect and the probability that the hidden state was the open state is either 1 or 0 (Figure 8.2, bottom panel). In a case where the magnitude of the difference between the open and closed states is not as large, our degree of confidence that the hidden state is open or closed is not as great and our posterior decoding therefore produces probabilities between 0 and 1 (Figure 8.4, bottom panel).

Of course, if one were modeling real data, one would wish to utilize a model that would produce the most insight into the nature of the data. Choosing models to apply to complex datasets can be as much art as science. In general, there is a trade-off between how well the model fits the data and model complexity. For Markov chains, the fit of the model to the data can be defined as how well the emission probabilities defined by the most likely path of states through the model (which is known as the Verterbi path; see [22]) match the data. For the high signal-noise data shown in Figure 8.2, there is essentially no model that could fit the data more closely than our two-state model. Our fit is essentially perfect in that in posterior decoding, we are certain we are either in the closed state (with a probability of 1) or the open state (with a probability of 1) and the actual amplitude and noise of each state are nearly identical to parameters that could be estimated from the large sample size represented by the many iterations shown in Figure 8.2. For the more difficult to model AQ5
situation of low signal to noise shown in Figure 8.4, we are no longer always certain in posterior decoding which state we are in and therefore no longer have a perfect fit of model to data. If we were to introduce a model with more states, we could improve the fit to the data but at the cost of arbitrarily increasing model complexity. To take an extreme example, if we defined a separate

[*] Technically, our model should also have initiation and termination with defined transition probabilities from these states to each other state in the model. See Ref. [22].

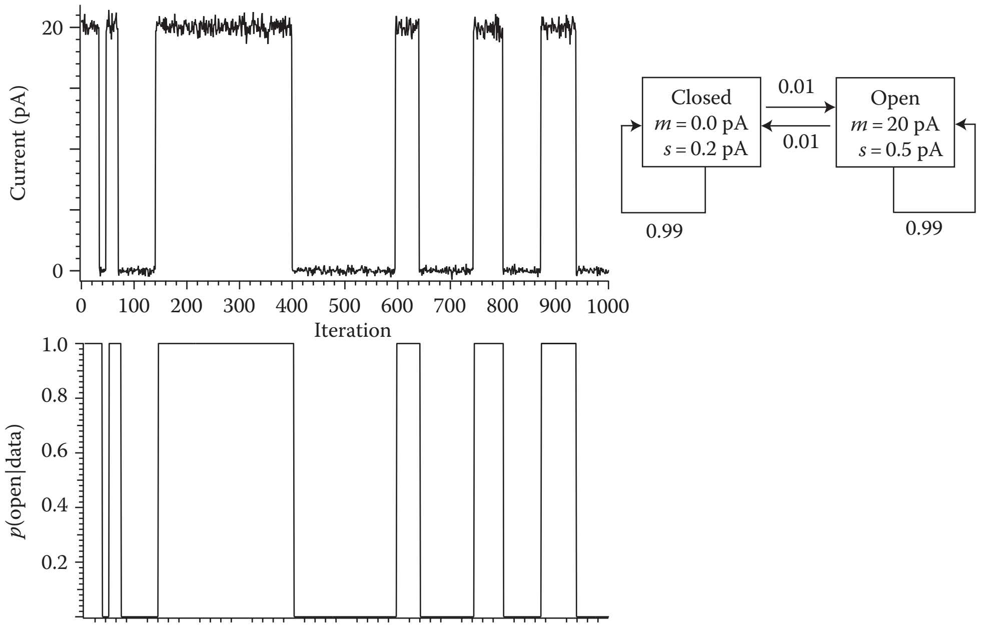


Figure 8.2 (top panel) Simulation of single-channel recordings from a two-state Markov chain; (bottom panel) given only the emissions and the underlying model, posterior decoding can determine the probability of being in the open or closed state for each iteration (time point). In this case, because the difference between the closed and the open state is large relative to the error, posterior decoding has no error or uncertainty and yields probabilities that are either zero or one.

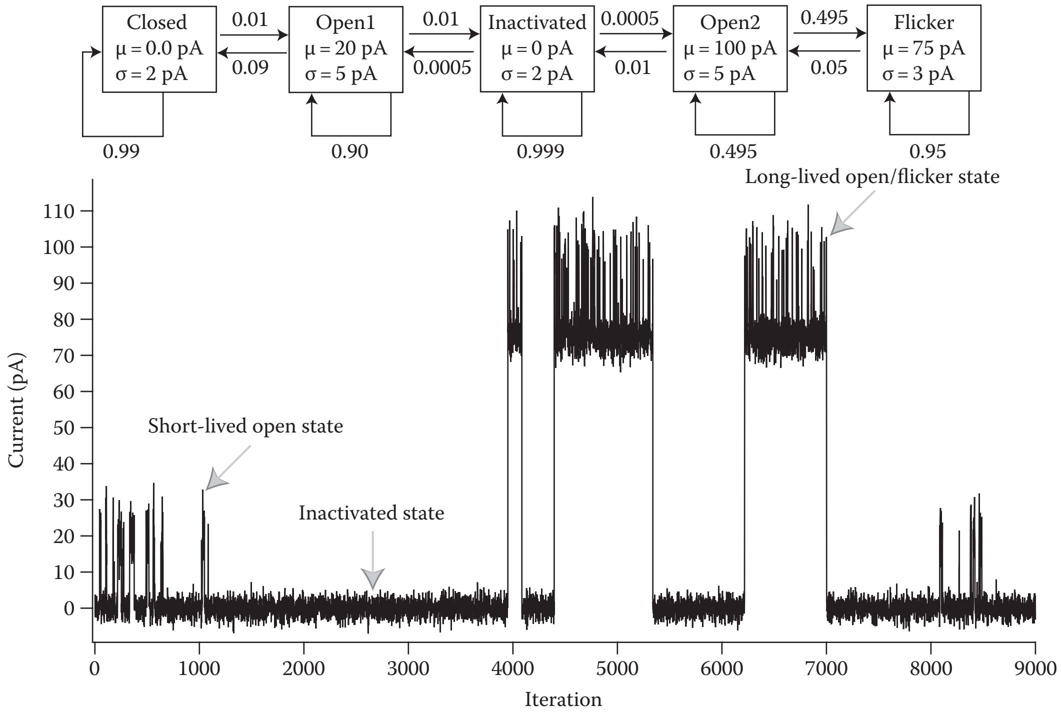


Figure 8.3 Simulations from a more complicated Markov chain that includes multiple open states, inactivation and flicker.

Markov state for every observed distinct amplitude, we could perfectly fit our data to this model in that with each possible amplitude given its own state, we would always know with perfect certainty in posterior decoding which state we were in and each state would perfectly match the observed amplitude at that iteration. This would, however, of course be an exercise in trivial overfitting that would not yield any insights into our data or any future predictive power. Our goal in finding the *best* model for a given dataset is to find the simplest model that explains most of our data. While there is an extensive literature devoted to this problem [33], there are no hard and fast rules that can determine when making a model more complex by adding a state produces a *better* model or simply leads to an overfit model. Ultimately, the most crucial test of an HMM or any other model constructed with statistical learning techniques is to be verified on a dataset that was in no way used for model construction.

AQ6

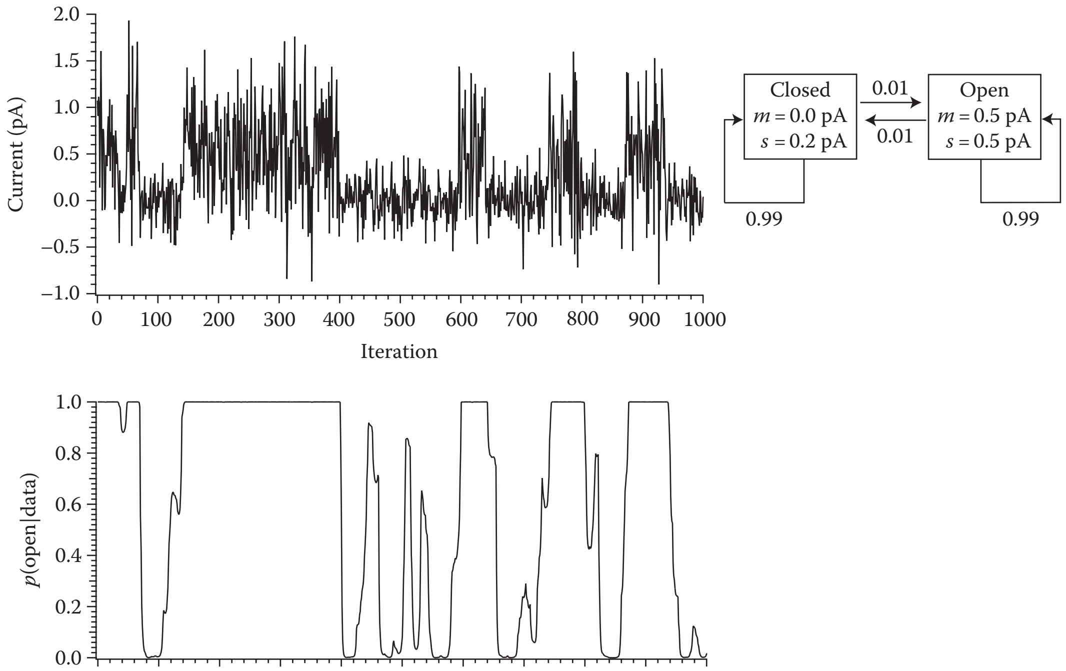


Figure 8.4 (top panel) Simulation of single-channel recordings from a two-state Markov chain with a smaller single-channel conductance than the simulations in Figure 8.2; (bottom panel) In this case, because the noise is as high as the single-channel amplitude, we cannot always tell when looking only at the emissions whether the underlying state was open or closed. Posterior decoding therefore produces probabilities other than zero and one.

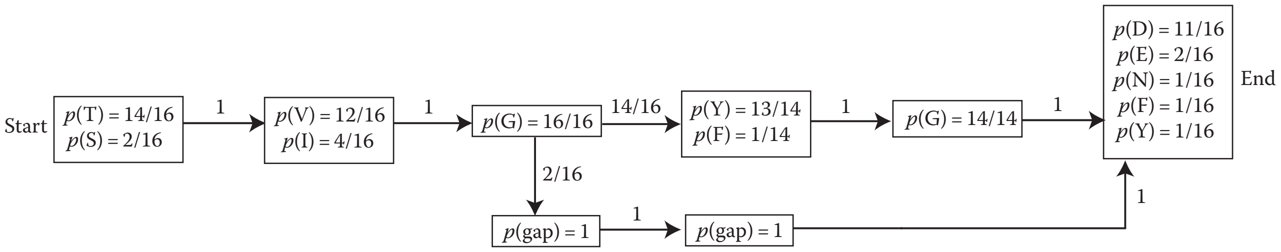
8.5 MARKOV CHAIN FORMALISM APPLIED TO HOMOLOGY DETECTION

One remarkable property about Markov models is that the same formalism and algorithms that allows for simulation and decoding of single ion channels can be applied to a wide variety of problems in bioinformatics. As an illustrative example, let us say that we wanted to build a simple profile model with which to detect the conserved pore sequence in potassium channels. We start by building a Markov chain based on strongly conserved positions as seen in a published [34] alignment of the pore region of K+ channels (covering the region TVGYGD in kcsa), which includes the more distantly related CNG channels (olCNG and rodCNG; see Figure 8.1 in [34]). Our highly simplified model of the six residue portion of the alignment could look something like this:

We have two kinds of states in our Markov chain. A *match* state emits residues with the frequency distribution derived from the residues in the column of the multiple sequence alignment. A *gap* state emits a gap in the query sequence relative to the alignment (with a frequency of 1). To calculate the probability of a particular sequence (say *T I G - - Y*) given our simple profile model, we start in the first state and record the probability of a T (14/16 since 14 of the residues in the alignment in [34] contain a *T* in the column corresponding to the first residue in our query sequence). We proceed to the second state (with a transition probability of 1) and record the probability of an I (4/16), then G in the next state (16/16). At this point, we make the transition to the gap state with a transition probability of (2/16). Once we are in the first gap state, our model requires a second gap (with a transition probability of 1) and finally in the last state, we recovered the probability of a Y (1/16). In general, the total probability of our path through the Markov chain is the product of all the emission and transition probabilities (see [22]). If we multiply all of these probabilities together, we achieve the $p(\text{sequence}|\text{model})$, which we read as the probability of the sequence given the underling model of the alignment. Of course, our simple model is not very flexible. For example, if a gap were not seen at the second position, or our sequence were to start with any other residue other than T or S, the resulting probability of $p(\text{sequence}|\text{model})$ would be zero. A more sophisticated model would realize that not every possibility is captured by the alignment that serves as our training set. Such a model would add the possibility of a gap occurring anywhere along the alignment and pseudocounts of residues to each column (some small probability for each residue that was not observed in the alignment column) to avoid zero probabilities. With such modifications, our model could produce a small, but nonzero probability, for any sequence fragment. We could then report a *log-odds score* for the significance our query sequence:

$$\ln\left(p(\text{sequence}\mid\text{model})/p(\text{sequence}\mid\text{random model})\right),$$

where the random model is based on the properties of some large collection of randomly generated unaligned sequences. As is the case for BLAST e-values, the significance of this score will be based on how well the random model captures the possible variation of all proteins that we might search. So, a random model based on membrane sequences (which, e.g., would have emission probabilities higher in hydrophobic residues) might



AQ7 **Figure 8.5** A highly simplified profile HMM capturing the alignment the *TVGYGD* region of the pore region of K+ channels shown in Figure 8.1. (From Doyle, D.A. et al., *Science*, 280, 69, 1998). Each state represents one column of the alignment.

yield a different probability than a general model based on all proteins. In general, because our random model might not always be appropriate for our query sequence, the log-odd scores, like BLAST e-values, should be considered only rough guides to the significance of a hit. In fact, it can be shown that using ion channels as query sequences, proteins can be assigned to incorrect families even when the log-odds scores are highly significant [35].

AQ8 The PFAM database (http://pfam.sanger.ac.uk/) attempts to build a profile HMM model that represents the alignment of every known protein family. The PFAM database takes an iterative approach to model building. First, a seed alignment is created from proteins known to share close homology. This alignment is used to construct a model, which is used to find more sequences, which in turn are used to build more refined models. The PFAM database provides aligned sequences for both seed and final alignments as well as representation of the HMMs that can be manipulated with the HMMer suite of software tools. Given a query sequence and a database of ion channels, represented as PFAM alignments, these tools can be used to ask whether the query sequence is represented by any of the models representing the ion channel.

Using the HMMer software and profiles built from the PFAM database, it has been demonstrated by comparing sequences to solved crystal structures that for a number of well-studied ion-channel families, the odds that prokaryotic and eukaryotic channels in fact have different protein folds are miniscule [35]. This validates a strategy of using prokaryotic structures to model eukaryotic channels. However, even within the context of a similar overall fold, small sequence variations can have enormous AQ9 functional consequences. For example, although CLC channels have a high degree of sequence homology that all but guarantees that they share a common fold [35], there is increasing evidence that some members of this family function not as chloride channels but rather as proton-chloride antiporters [36]. These sorts of functional differences, while not currently predictable from just sequence information, presumably reflect small-scale differences in structure and energetics, which nonetheless must AQ10 have enormous consequences for phenotype (Figure 8.5).

8.6 CONCLUSION

Electrophysiological recordings offer perhaps the most detailed measurements of real-time protein energetics and small conformational changes available for any protein. As described earlier, the advent of new sequencing technology combined with informatics tool is producing an avalanche of information on how sequence can vary within populations and across phylogenetic space. A central open question is whether we can use sequence variation to make a priori predictions about biophysical differences in ion-channel function or, instead, will determination of ion-channel function always be a strictly empirical matter in which the effects of sequence variation can be known only from experimental data.

In their influential review [11], Manolio et al. note that GWAS studies have focused on common alleles and have relied "almost exclusively on statistical evidence." They argue that if the missing heritability is to be found in rare variants then "the challenges of sifting through the millions of rare variants in which two individuals differ" may be substantial and may require a "return to biology if rare variants are to be grouped and analysed properly." If missing heritability is due to gene–gene [14] or gene–environment [15,16] interactions, the number of possible hypotheses that will need to be evaluated by future genomics studies likewise borders on the infinite. If the future progress of genomics indeed requires a *return to biology* to prioritize these many possible interactions, the study of ion channels will likely be front and center. Not only are ion channels crucially involved in most major human diseases, but electrophysiological techniques allows for exquisite linking of ion-channel structure and function. The merging of genomics techniques with tried-and-true experimental measurements points to a future in which ion-channel sequence variation captured from genomics is placed in a rich biological context encompassing both the biophysics of channel function and the important role that channels play in health and disease phenotypes.

ACKNOWLEDGMENTS

We thank Rick Aldrich and Zheng Chang Su for helpful comments on a previous draft of this chapter.

REFERENCES

1. Hodgkin, A.L. and A.F. Huxley, A quantitative description of membrane current and its application to conduction and excitation in nerve. *J Physiol*, 1952. **117**(4): 500–544.
2. Ho, K. et al., Cloning and expression of an inwardly rectifying ATP-regulated potassium channel. *Nature*, 1993. **362**(6415): 31–38.
3. Schrempf, H. et al., A prokaryotic potassium ion channel with two predicted transmembrane segments from Streptomyces lividans. *EMBO J*, 1995. **14**(21): 5170–5178.
4. Tamkun, M.M. et al., Molecular cloning and characterization of two voltage-gated K+ channel cDNAs from human ventricle. *FASEB J*, 1991. **5**(3): 331–337.
5. Tempel, B.L., Y.N. Jan, and L.Y. Jan, Cloning of a probable potassium channel gene from mouse brain. *Nature*, 1988. **332**(6167): 837–839.

6. Dascal, N. and I. Lotan, Expression of exogenous ion channels and neurotransmitter receptors in RNA-injected *Xenopus* Oocytes, in *Protocols in Molecular Neurobiology*, A. Longstaff and P. Revest (Eds.), 1993, Springer, New York. pp. 205–225.
7. Hoshi, T., W. Zagotta, and R. Aldrich, Biophysical and molecular mechanisms of Shaker potassium channel inactivation. *Science*, 1990. **250** (4980): 533–538.
8. Cheng, S.H. et al., Defective intracellular transport and processing of CFTR is the molecular basis of most cystic fibrosis. *Cell*, 1990. **63**(4): 827–834.
9. Gordan, S.E. and W.N. Zagotta, Localization of regions affecting an allosteric transition in cyclic nucleotide-activated channels. *Neuron*, 1995. **14**(4): 857–864.
10. Fodor, A.A., S.E. Gordon, and W.N. Zagotta, Mechanism of tetracaine block of cyclic nucleotide-gated channels. *J Gen Physiol*, 1997. **109**(1): 3–14.
11. Manolio, T.A. et al., Finding the missing heritability of complex diseases. *Nature*, 2009. **461**(7265): 747–753.

AQ11 12. Consortium, C.-D.G.o.t.P.G., Identification of risk loci with shared effects on five major psychiatric disorders: A genome-wide analysis. *The Lancet*, 2013. **381**(9875): 1371–1379.

13. Hindorff, L.A. et al., Potential etiologic and functional implications of genome-wide association loci for human diseases and traits. *Proc Nat Acad Sci*, 2009. **106**(23): 9362–9367.

AQ12 14. Zuk, O. et al., The mystery of missing heritability: Genetic interactions create phantom heritability. *Proc Nat Acad Sci*, 2012.

15. Murcray, C.E., J.P. Lewinger, and W.J. Gauderman, Gene-environment interaction in genome-wide association studies. *Am J Epidemiol*, 2009. **169**(2): 219–226.
16. Parks, B.W. et al., Genetic control of obesity and gut microbiota composition in response to high-fat, high-sucrose diet in mice. *Cell Metab*, 2013. **17**(1): 141–152.
17. Rusch, D.B. et al., The Sorcerer II global ocean sampling expedition: Northwest Atlantic through Eastern Tropical Pacific. *PLoS Biol*, 2007. **5**(3): e77.
18. Yooseph, S. et al., The Sorcerer II Global Ocean sampling expedition: Expanding the universe of protein families. *PLoS Biol*, 2007. **5**(3): e16.
19. Altschul, S.F. et al., Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucl Acids Res*, 1997. **25**(17): 3389–3402.
20. Eddy, S.R., Where did the BLOSUM62 alignment score matrix come from? *Nat Biotech*, 2004. **22**(8): 1035–1036.
21. Ng, P.C., J.G. Henikoff, and S. Henikoff, PHAT: A transmembrane-specific substitution matrix. *Bioinformatics*, 2000. **16**(9): 760–766.
22. Durbin, R. et al., *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. 1998: Cambridge University Press. AQ13
23. Needleman, S.B. and C.D. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, 1970. **48**(3): 443–453.
24. Smith, T.F. and M.S. Waterman, Identification of common molecular subsequences. *J Mol Biol*, 1981. **147**(1): 195–197.
25. Kent, W., BLAT—The BLAST-like alignment tool. *Genome Res*, 2002. **12**(4): 656–664.
26. Korf, I., BLAST. 2003: O'Reilly Media. AQ14
27. Eddy, S.R., Profile hidden Markov models. *Bioinformatics*, 1998. **14**(9): 755–763.
28. Bateman, A. et al., The Pfam protein families database. *Nucl Acids Res*, 2002. **30**(1): 276–280.
29. Bateman, A. et al., The Pfam protein families database. *Nucl Acids Res*, 2004. **32**(suppl 1): D138–D141.
30. Qin, F., A. Auerbach, and F. Sachs, A direct optimization approach to hidden Markov modeling for single channel kinetics. *Biophys J*, 2000. **79**(4): 1915–1927.
31. Qin, F., A. Auerbach, and F. Sachs, Hidden markov modeling for single channel kinetics with filtering and correlated noise. *Biophys J*, 2000. **79**(4): 1928–1944.
32. Venkataramanan, L. and F.J. Sigworth, Applying hidden Markov models to the analysis of single ion channel activity. *Biophys J*, 2002. **82**(4): 1930–1942.
33. Hastie, T., R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Second Edition. 2009. AQ15
34. Doyle, D.A. et al., The structure of the potassium channel: Molecular basis of K+ conduction and selectivity. *Science*, 1998. **280**(5360): 69–77.
35. Fodor, A.A. and R.W. Aldrich, Statistical limits to the identification of ion channel domains by sequence similarity. *J Gen Physiol*, 2006. **127**(6): 755–766.
36. Matulef, K. and M. Maduke, The CLC 'chloride channel' family: Revelations from prokaryotes (Review). *Mol Membr Biol*, 2007. **24**(5–6): 342–350.

AUTHOR QUERIES

[AQ1] Please check if the edit to the sentence starting “The e-values can be thought ...” conveys the intended meaning.
[AQ2] Please provide the author name, article title and year of publication for URL address throughout the chapter.
[AQ3] Please check the equation for correctness in sentence starting with “We can choose....”
[AQ4] Please check the sentence starting “the μ and ...” for sense.
[AQ5] Please check the sentence starting “For the more difficult ...” for sense.
[AQ6] Please check sentence starting “While there is an ...” for sense.
[AQ7] Please check figure caption 8.5 for sense.
[AQ8] Please provide the expansions for “CFTR, PFAM, TVGYGD, kcsa, HMMer, and CLC” at the first mention, if appropriate.
[AQ9] Please check the sentence starting “For example, although CLC ...” for sense.
[AQ10] Please check the insertion of Figure 8.5 for correctness.
[AQ11] Please check the insertion of volume, issue and page range for Ref. [12].
[AQ12] Please provide volume number, issue number and page range for Ref. [14].
[AQ13] Please provide publisher location for Ref. [22].
[AQ14] Please provide complete details for Ref. [26].
[AQ15] Please provide publisher name and location for Ref. [33].