yes these techniques are very applicable to clinical data.  As I said, these HMM things are applicable to the sleep studies, to the Pregnancy studies, to Armen and Mridu's heartbeat things, Nicole's data, basically all the clinical data we currently have.  It might not be clear to you how Nicole's data can be fit into an HMM, but it can.  Remember that because of the notion of state, the solution of a differential equation is indeed a Markov chain.  And as I mentioned yesterday, our observations will be a noisy version of that.  So basically all the data that we have in the lab is an HMM.  And learning about Markov chains is very important to learn about control, which is the next topic i want to train people on.  As for the application to clcinical data, I thought I explained to you before that the HMM is core to basically all the data we have so I wanted to sequentially move with you through it, in increasing complexity (and increasing relevance to data we have).  In case I wasn't clear or you forgot, here is the plan, rationale, and applicability to clinical data:

* **Theory**: discrete X, discrete Y.  play God and generate samples from known stats.  Implement filter, smoother, MAP sequence estimation.  **Stats**: ML estimation via EM algorithm.  **Application**: gene sequencing shit from CAARMS.  **Your Status**: Playing God is DONE.  Filter is DONE. Other stuff: smoother/MAP seq estimation/ML  estimation via EM alg: we can come back to this - I don't want to overwhelm you with moving in too many directions so we will return to this.  Good news is that once you learn the smoother/MAP seq estimation/EM alg once, basically the same pseudo-code will applies to all the other bullet points
* **Theory**: discrete X, cts Y.  filter, smoother, MAP.  **Stats**: ML estimation via EM algorithm.  **Application**: sleep EEG data and seizure EEG data.  **Your Status**: Playing God is DONE (I think, please confirm?) Filter is almost DONE (you just need to do Bayes rule).
* **Theory**: continuous X, discrete/cts Y.  filter, smoother, MAP sequence estimation.  **Stats**: ML estimation via EM algorithm.  **Application**: behavioral modeling with Kinect/LEAP, joint behavioral and physiological modeling like Emery Brown paper you attached and my postdoc work(*they are mathematically the exact same conceptual framework*), adult heartbeat/heart-rate-variability, Sandy Ramos fetal heart rate, MJ/Lance Prince neonate heartbeat/respiration/heart-rate variability stuff., **Your Status**: filter is conceptually the same except sums becomes integrals.  Practically, implementing it in Matlab is"harder" because
  + X is continuous and so the one-step prediction update becomes no longer a sum but now an integral.  In general you cannot do the integral so there are other approaches people use.  Many people model X with Gaussian Markov statistics and the noisy channel from X to Y as additive Gaussian.  In such cases, you can implement the Kalman filter (which I can train you on).  More generally, there are approximation methods that people develop (including the stuff Emery did in the paper you mention above).  Me and Sanggyun's optimal transport shit can also be used here but I don't want to introduce this to you just yet
  + X is continuous so the Bayes rule is also harder, the denominator is an integral.  Here, again, me and Sanggyun's optimal transport shit can solve this generally.  Again, let's not worry about this yet.
* I did not include the Nicole scenario above because it is easier than all of these: note that given the initial condition, X is a non-stochastic Markov chain, it has deterministic evolution.  So this is easier. ML estimation can be done on Gaussians using BENG 100.
* soon, in parallel, I will be training people on stochastic control theory and inverse optimal control (with many applications to modeling behavioral data).  All that needs to be known is Markov chains and new shit that I will teach on dynamic programming and Bellman's equation (same idea actually as the MAP sequence estimation: shortest path algorithm).  All this is in my notes on stochastic control - the way I make the connection is from Markov chains to control.  So perhaps after you finish the discrete X cts Y filter problem, I can then train on MAP sequence estimation because it makes the connection to control.

so yes, I understand you want to move faster towards applying this to various clinical data - I want you to do that too :).  But I think we have to sequentially go through it with increasing conceptual/algorithmic difficulty so that it is clear you understand, and the rate limiting step will be how fast you can pivot from one bullet point to another, with a level of understanding that I feel comfortable with.   So the punchline is that "*I can show you the well, but I can't force you to drink the water*". If you want to move faster, I think we will need for you to improve on how you walk away from our previous discussion and independently try to make progress and conceptually understand how what you are doing now relates to what you did before, what knobs are the same, and what knobs are changing.  (*For example: playing God for discrete X and then the analogous playing God for cts X, calling "rand" gets changed to calling "randn".  You don't use densities here, you perform transformations on randomly generated samples*).  Also, I will need for you to get comfortable independently going back to the BENG100 notes and/or HW to fill in details.  When I say something like "this is very similar to HW 3 problem 2", I am implicitly saying that it might behoove you to read through those associated notes, look at that hw problem, and the carefully-written up solution, and see how this relates to your problem and how you might be able to implement things in light of this.  
  
Now I know you are very very busy doing many things and **I cannot thank you enough** for that (*by the way I am nominating YOU for the postdoc mentoring award - you by far have that on LOCK amongst all the postdocs*).  So I was not at all wanting to move faster - you have alot of shit going on.  But when I read your email, I got the idea that you want to move faster.  All I am saying is that, if you want to independently move faster on it and apply to the clinical data, remember that "*you can't get something for nothing*".  To really understand this stuff and independently use it, it will take blood sweat and tears and a strong attempt at getting your hands dirty, independently tackling this.  The key thing that is important to do is to understand what is happening conceptually and then understanding what knob is changing and what that means in terms of the detailed math and/or what is happening in terms of changes in the code.  Once you get comfortable with this, pivoting from one of these bullet points to the other should begin to come easier and you will be able to map all this stuff to different datasets with increasing agility.  
  
Sound good?  Hope this email helps.  I think you are doing a great job with managing everything in the lab and I cannot thank you enough.  If you want to run faster, then I think YOU will have to run faster - that is all I am saying :).   In fairness to you, maybe this picture that I created with these bullet points, connections to data, etc was clear in my head and I was saying "trust me - follow this path and you will get there".  And maybe I should have provided this glimpse of how I am seeing this training, how it will relate to clinical applications, etc.  So for that, I apologize.   If you like, you can turn all these bulletpoints, etc into a ppt for a training manual (i was working on one for you when i was traveling but I got tied up - you should see a draft in dropbox).  We can use this training manual for everyone who wants to do HMM/control analytics.    
  
big smile,  
-t