

How to get started in computational neuroscience? - NeuroTechX Content Lab - Medium

NeuroTechX Content Lab

28-36 minutes

Two domain experts from academia and industry offer top-shelf directions.



A world with computational neuroscience is better than a world without it. We can measure. We've got data. How do we figure out what means what? The brain is a complex organ but it is finitely complex. Computational neuroscience is key to cracking that complexity. Let's ask a pair of experienced practitioners about how to join this code-cracking enterprise.





The representative of young academia is Artem Kirsanov. Artem is a student of computational neuroscience and biophysics, and a researcher at Moscow State University. He's also [an exquisite Youtuber](#).

Research-wise, Artem works in the Laboratory of Extrasynaptic signaling, led by Dr. Alexey Semyanov. The lab's main body of work is focused on (1) astrocytes, particularly on the analysis of calcium waves, which are likely to play a role in the modulation of synaptic activity through the release of gliotransmitters. ([Recent opinion paper](#).) And (2) on the analysis of hippocampal theta activity from freely moving mice.

At the moment, Artem is building computational models for (1) astrocytes in silico and analyzing their calcium dynamics, and (2) models for theta modulation.



The industry delegate is Cesar Echavarria. Cesar works as a neuro data scientist at [Synchron](#) — the company that makes the now well-known endovascular stentrode. He's an MIT and Harvard-trained neuroscientist with [an abundance of research experience](#). Cesar's PhD project focused on the cortical computations that give rise to visual perception.

Incidentally, his dissertation advisor was David Cox, who leads a 3-part online course provided by HarvardX called [Fundamentals of](#)

[Neuroscience.](#)

As an output of our conversation, Artem has released a video on the topic. Below it follows a complete Q&A with both Cesar and Artem. Their answers were lightly edited for clarity.

Let's start with a few fundamental questions. How can computers help us understand how the nervous system works?

Artem:

I guess, it can be very loosely divided into 2 broad groups — data analysis and simulations.

First of all, there is an overwhelming amount of data we collect from the real brain with experimental techniques. And computers are life-saving tools for dealing with these huge amounts of information at every stage of the analysis.

For example, let's say you have a video recording of calcium dynamics unfolding in an astrocyte (a type of glial cell in the brain). Before you can do something with this data, it surely needs preprocessing — like denoising, filtering, cropping, etc. Then, depending on the design of the experiment and the hypothesis you're testing, you start to extract some useful information from this data — writing scripts to describe the calcium patterns and linking them to behaviour / diet / physiological state / etc. Then you usually want to perform statistical tests to establish the significance of your results and create plots and other visualisations.

This is somewhat a simplified example, but it illustrates the backbone of the data analysis process. Computers make every stage of it possible, from data collection to presentation of the results.

But computers can also serve as a **source of data** of their own. When we want to understand how something works, we often want to build a simplified model of this phenomenon, in an attempt to explain it. And computers are what we use to build and validate the models. Detailed numerical simulations can help understand mechanisms behind brain function and test hypotheses. Results from the simulations can be then compared with real-world data, to test whether the model is consistent with reality.

Of course, these two approaches should be used in synergy. When you collect the data, find an interesting phenomenon and propose an explanation, it is really valuable to build a model to test this very hypothesis as additional evidence. Likewise, when you start with the model and discover intriguing behaviour, it's a good idea to run an experiment to see whether something similar occurs in a real brain.

Cesar:

Interpretation 1: Not very well. Wanting to understand the nervous system by studying computers is analogous to studying planes to understand bird flight. A few core principles are at work, but they're vastly different systems.

Interpretation 2: Computers are an essential tool to help scientists analyze data and conduct simulations. They are by no means a requirement, but they speed things along and make a lot of analyses possible.

From your perspective, what's good about computational neuroscience? What's bad?

Artem:

As I see it, the greatest advantage of computational neuroscience is that building models and using fancy data analysis methods can allow us to go **beyond the current experimental limitations** and get us access to variables that are normally latent and can't be measured directly.

For example, if you want to investigate the role of cell morphology (its shape) in determining calcium signaling, you can make experimental measurements and search for patterns.

Complementary to that, it could be helpful to build a model to twist and turn the simulated cell geometry in obscure ways, to see how it affects the calcium levels. Probing the model in such fashion can lead to a discovery of unexpected emergent phenomena that would otherwise remain unknown.

But this can also be its biggest weakness. It could be tempting to just bury yourself in the simulation world and study the artificial system for its own sake, with no connection to reality. That's why I

think it's important not to turn away from experimental data, and not to forget what you set out to study in the first place.

Cesar:

I think any effort to increase our understanding of the natural world is fundamentally good. What can be misguided are efforts to understand the nervous system with only the tools of computational neuroscience (modeling, data analysis, and such). The nervous system is very complex and a good understanding of its structure and function often comes about by considering and incorporating knowledge across various levels (e.g. genetics, cell physiology, behavior).

What are some high-impact problems the field of computational neuroscience should solve in the short to medium term so that the discipline can move forward?

Artem:

That's a tough question! Personally, as a "semi-experimentalist"* I'd say that most limitations come from the experimental side of things, which are closely tied to computational neuroscience. **(My work involves a combination of experimental and computational methods. I.e. I perform surgeries on mice to implant electrodes, record brain activity, collect the data, and then load that into the computer to process.)*

For example, we have the capabilities to run detailed simulations of thousands of artificial neurons and access the value of membrane potential at every point of the cell. But we are still at nascent stages when it comes to the technology to reliably do this with a real neuron. The same is true for almost every aspect of the brain — from molecules, through cells and circuits to behaviour and large-scale connectivity.

When you are building a model, you are limited only by your imagination and computer power. But the validity of it is questionable if there is no way to compare your results with the data acquired from the real brain.

Cesar:

The last 10–15 years have seen the development of tools that let us collect a vast amount of neural data (many animals, many neurons, many days worth of observations). To make the most of these data and increase our knowledge about the nervous system, more work is needed on two fronts:

1. **Developing analysis tools to extract meaningful information from these data.** Simply scaling up the methods that have been used to study single neurons is not enough. We need to use methods that capture the emergent properties of large populations of neurons.
2. **Developing tasks that are complex and naturalistic.** Simple tasks produce simple neuronal activity. The richness of our nervous system did not develop to solve simple two-choice tasks. Rich behavior is necessary to capture and understand the computational abilities of the nervous system.

If I'm interested in computational neuroscience, what should be my first step? What math do I need to know?

Artem:

I'm a big believer in **learning by doing**. During my freshman year of college, I had the impression that I needed to first learn all the necessary math (such as calculus, linear algebra, differential equations, etc.) **before** I had the moral right to email the professor inside the lab and ask about taking part in their research.

But later I realised that **you can't prepare for and learn everything in advance**. The best way to learn is to dive right in, by doing the work and getting your hands dirty. That's why I think you shouldn't stress too much about the math. It will gradually come to you as you learn *on the go*, in parallel to applying the theory to solving real problems.

Additionally, the exact mathematical toolset will be determined by the particular project you're working on. For instance, if you're interested in connectivity and network neuroscience, knowledge of graph theory is essential. Meanwhile, if you are drawn to building

realistic computational models of neurons, you'd probably want to put more emphasis on dynamical systems and biophysics. Of course, some topics are ubiquitous throughout the majority of problems. Elementary knowledge of linear algebra (such as matrix multiplication) will certainly come in handy no matter what you do.

That's why I think the first step should be to **start doing projects as soon as possible**. If at any point you find yourself lacking some necessary knowledge, take some time to familiarise yourself with the main concepts and ideas. This kind of learning is more efficient because you're not absorbing information "in a vacuum," but you learn it in the context of the problem you're trying to solve.

This can mean starting the practice either on your own, or searching for opportunities to work as a part of a research lab at your local university.

Cesar:

Computational neuroscience is a very big topic with different mathematical models describing phenomena at various scales. On one end of the spectrum, these models describe [how specific ion-channels contribute to the electrical properties of neurons](#). On the other end of the spectrum, these models describe [how individual fish in a group coordinate their movement with their neighbors to avoid predators](#). In between, you find descriptions of [how a population of neurons integrates information to guide an animal's behavior](#).

My advice is to explore the space and find a topic and phenomenon that piques your interest (motor control, sensory perception, memory, etc.). Once you find an interesting thread, just start pulling on it.

You need a good knowledge of calculus and linear algebra to really understand most research in computational neuroscience these days. You should also have a good understanding of statistics in order to properly analyze and interpret experimental data of any sort (neurophysiology experiments or computer simulations).

What software do I need to know, and what programming languages can I use?

Artem:

From my experience, the two most popular choices for programming languages are Python and MATLAB. When you join a lab, it will very likely already have some “*inner culture*” established — like custom-written modules and a data processing workflow. In this case, it’s best to fit in and use what they’re using, at least in the beginning.

For self-studying, when you are on your own, you can choose whichever you want. What really matters is the skill of constructing algorithms, while the exact language you’re using to implement them is not really important. Personally, I went with Python because it’s free, while MATLAB is a proprietary software, which could be costly to obtain if you’re not affiliated with a university that has the license. Additionally, I believe Python is more popular and there are a ton of packages created for doing almost everything. There is also a programming language called Julia, which is under active development, aimed at scientific computing. It is very promising, but it’s very young and not as widespread as Python yet. While it’s certainly worth checking it out and seeing what Julia can do, if you’re choosing between the two, the safest bet is to go with Python, in my opinion.

Cesar:

For industry, Python is a must. Specifically, you should be comfortable using NumPy, pandas, scikit-learn. For deep learning, PyTorch and TensorFlow are equally good. MATLAB and R are popular in academia and some interesting packages are released with these languages so being familiar with these languages is definitely useful.

Can you recommend any online tutorials?

Artem:

As for tutorials on Python per se, I’m not really familiar with much of what’s out there. The thing is, I learned to code back in middle-high school with a teacher, who coached our team for competitive programming. But I’m sure there is a huge variety of educational resources to get started with Python from scratch. The general

advice I can give though is to **put more emphasis on solving problems that are not straightforward and involve algorithmic thinking** rather than plain syntax practice. The majority of such problems come from different programming olympiads. You can find them on codeforces.com. I know it can seem scary, but this is by far the best way to learn, in my experience.

As for Python in the context of computational neuroscience, as an entry point I can recommend a fantastic online textbook by researchers from EPFL, titled [Neuronal Dynamics](#). It contains the textbook itself, video lectures, and most importantly, Python exercises.

Cesar:

The computational neuroscience and deep learning courses by [Neuromatch Academy](#) are the finest out there. The [Neural Data Science course](#) taught by Bradley Voytek is a good starting point to learn about the analysis of various types of neurophysiological data.

Lina (editor):

Editor's recommendation: [IBRO](#)'s Computational Neuroscience Schools are one of the best opportunities to gain knowledge and meet people interested in the topic. Attendance can be online or in-person, in fact, there is one school where applications are still open — [Computational Neuroscience, Neurotechnology and Neuro-inspired AI Autumn School](#). [NCAN](#) also offers a summer course from time to time, as well as an internship in neurotechnology.

How do I find project ideas?

Artem:

It could be tough to come up with a project when you're on your own. Honestly, I don't have a complete answer at this point. But I think it's vital to follow 2 rules when you are choosing (or thinking up) a project.

First of all, **make sure that it's something you're personally resonating with**. If somebody recommends a topic, but it doesn't evoke any sparks of excitement, I think it's best to keep on

searching for something else. This helps to make sure you're fully invested in following the project to its completion. It's especially relevant when there are no external motivators for you to pursue it.

Another thing to keep in mind is *the Goldilocks rule* — the project should not be too difficult for your current level (otherwise you will soon get discouraged) or too easy (it has the risk of feeling boring and pointless). The sweet spot is right at the edge of your abilities. Hard enough for you to feel the sense of accomplishment and excitement when you complete some intermediate milestones, but not too hard to make you abandon it because you don't make any progress.

As a possible suggestion, when you are reading papers look for some aspects of data analysis / modeling that excite you. If something resonates with you, you can set up a mini-project for yourself to first make sense of the authors' code (usually available on something like GitHub) and then create a simplified version of it on your own. For instance, if this is a paper that contains some computational model of rhythm generation, check out their code, what modules they are using, and what's the big picture of their approach. Then you can go ahead and try to create a smaller model with similar methods and see if it generates some rhythmic behaviour.

It's not ideal, but it gives you an opportunity to practice on real problems that are currently being solved by research groups.

Cesar:

Two good ways to find ideas:

1. **Read papers.** Reviews are particularly useful to get a good context on the state of the field on a given topic. See what questions linger on your mind and think about how you would go about answering them.
2. Hop on **neuroscience Twitter**. Twitter is a good public forum to get familiar with and engage in discussions about various scientific ideas. These discussions can provide food for thought and ideas on potential projects. Some active and interesting people to follow: [@NicoleCRust](#), [@tyrell_turing](#), [@TrackingActions](#), [@KordingLab](#), [@nschawor](#), [@patrickmineault](#), [@dlevenstein](#). Though these are biased to my own interests 😊

For any project idea, I would recommend starting without concerns about the logistics of finding the appropriate data. **Think about the ideal experiment** then gradually adjust your original idea as you figure out logistics. You will likely end up with an imperfect or limited answer to your original questions. That's okay. Part of the process is to understand the limitations of your experiment due to logistics or the current state of technology.

Where can I find data to play with? How do I then tell a good dataset from a bad one? What about collecting my own data?

Artem:

Luckily, nowadays there are lots of datasets freely available online. You can use specialised resources, like openneuro.org or neuromorpho.org. As an alternative, you can always email the authors of a particular paper you like and ask for their data (don't feel shy to do it — it's a totally normal practice).

I would say that the quality of the dataset has different meanings, depending on the goal of your project. For example, if you are analysing the morphology and branching of dendrites, then a dataset of neuron reconstructions, which has dendrites perfectly labeled, but some axons or somas (cell bodies) are missing, would be fine. If, on the other hand, you're focusing on axon morphology, the very same dataset wouldn't be a great choice.

But quite often, the best way is to collect your own data. That way you are not limited to somebody else's experimental design and conditions. Instead, you can intentionally design an experiment to test your particular unique hypothesis, which is much more valuable. It goes without saying that collecting good data is only possible when you are a part of a research lab, which has all the necessary equipment and personnel.

That's why when you are only starting on your own, you will probably have to use open datasets and models. But don't get discouraged! At this stage in neuroscience, we are generating more data than we can meaningfully process and interpret. There are even entire research groups, who process the data collected somewhere else.

Cesar:

The [Collaborative Research in Computational Neuroscience database](#) and [figshare](#) are popular data repositories with a wide variety of groups. Google also has a [datasets search site](#). One issue with the datasets you find in these databases is that the organization of the data varies across datasets. This creates a bit of a burden on you to do some data-wrangling before being able to start on your research project. For this reason, I'm a fan of the [Neurodata Without Borders](#) (NWB) data standard for neurophysiology data. The [DANDI archive](#) is a growing data repository as more labs adopt and publish datasets with the NWB standard.

Once you get started with a dataset, you should perform some sanity checks with the data to make sure that the data is of good quality and useful for your analysis. The specific sanity check will depend on the data set. For example, for a dataset that contains neural activity from the primary visual cortex, plotting tuning curves to stimulus orientation is a good idea. Some pre-processing might be required to get a good signal from the data so you should also refer to the publications that used these datasets for these details.

All models are wrong — what matters is to what degree. How can I assess the quality of a model, i.e. how well it reflects reality?

Artem:

That is indeed correct! Any model inevitably simplifies reality. First of all, I'd say that choosing the right model is an important step, and it is actually a form of art. The quality of a model depends on the particular question you are asking. Since our computational resources are finite, it's important to **find the right balance between what to include and what to neglect**. A good model should be complex just enough so that it allows you to test the hypothesis in the first place. But not more complex than it is necessary.

For example, if you are modeling the dynamics of voltage-gated ion channels and their effects on the shape of action potentials, you'd have to put emphasis on the biophysical properties of the channel,

and set up a handful of equations for the kinetics of its subunits, etc.

On the other hand, suppose you are designing a computational model to investigate how the connectivity of the network affects its ability to synchronise the action potentials. In this case, to make the simulation of a large network possible, you will probably want to use a very simplified point-neuron model with just a couple of variables. Because the focus is now on the large-scale behavior of the system (namely, **timings** of action potentials), we can ignore lower-level details which are not in the scope of the question (like explicit kinetics of ion channels and the shape of the spike).

Cesar:

To assess the quality of the model, you need data from an experiment that seeks to test a prediction that the model makes about the structure or function of the nervous system. The data will reflect factors that are within and outside your control so a thoughtful experimental design and analysis are necessary. If the data does not match the model then you revise the model or consider an alternative model. If the model's predictions are validated by the data then you go on to test another prediction and so the dance continues...

Starting on your own can be hard and make people prone to loss of motivation. How can I learn, or work on a project, with other people?

Artem:

You can find like-minded people on Reddit or Twitter, talk about each other's interests and maybe organise something like regular journal clubs or study group meetings, where you will meet (either in person or virtually) and discuss relevant papers or contents of a textbook. The same goes for working on a project together.

Since I joined the lab quite soon after starting on my own, I haven't tried this in the context of neuroscience. But back in high school we would occasionally meet with a group of friends in a coffee shop and work on math problems together.

This is a nice technique, but the effectiveness varies for different

people. For example, I personally prefer intrinsic motivation and I love to work mostly independently. But if you need someone to work together or hold you accountable, definitely go for it!

Cesar:

The [Neuromatch Academy](#) summer courses are a good way to stay motivated because you get to learn with people with similar interests at the same time. Additionally, groups of students undertake a research project with a published dataset. This can serve as a beginning of a longer-term project if you find a topic and group that is a good match for you.

Sophie (editor):

Editor's recommendation: [NeuroTechX](#) is another great global community for meeting like-minded neurohackers, from students to experts and academia to industry!

Do I need a mentor? What's the best way to find one in this field?

Artem:

Having a mentor definitely helps, because an experienced person can guide you and point you in the right direction. In a conventional scenario, when you join a lab as a student or a postdoc, you automatically get mentors — there is always a PI and senior staff to supervise you and help you on that path.

Lack of continuous mentorship is certainly one of the pitfalls of starting on your own, and I don't have a great solution to this as of right now. If you have any particular questions you can always email competent people, like PIs and senior researchers. There is a good chance that you will get a reply.

But persistently mentoring someone requires a significant amount of time on a regular basis, and not a lot of people are willing to make that sacrifice to a complete stranger. This is why, in my opinion, it's best not to stay on your own for too long and join a lab or a research group.

Cesar:

A mentor is always good. I recommend finding someone who shares common interests at your academic institution or company. You can also always reach out to people on Twitter, LinkedIn, or the [NeuroTechX](#) Slack. It may be hard to find a longer-term mentor in this way, but people are generally open to having a virtual or in-person coffee chat and answering any questions you may have (myself included).

What's the best way to advance after I learn the basics?

Artem:

Depends on your end goal. If you wish to follow a “conventional” path of publishing papers, writing grant proposals, and moving up the ladder, it’s almost necessary to join a lab at some point, the sooner — the better. There are many research groups who routinely collect experimental data, but lack people who could do data analysis. Luckily, being a computational neuroscientist also means that you have the capacity to do work remotely.

If you are learning for other purposes, such as personal interest or science communication, the next steps will vary drastically.

Cesar:

The answer depends on the rabbit hole you find yourself in. My general advice is to keep up to date with the latest research on your topic of interest. You can set up email alerts from Google Scholar for certain keywords. Expanding your Twitter network is also a good way to stay on the cutting edge because people will often post good summaries of their latest publication with tweet-prints.

What are the main differences between working in academia and working in the industry?

Cesar:

Two big differences come to mind:

1. In **academic research**, you are usually involved in the whole lifetime of a project: from cradle to grave. In industry, you are often dropped into work that has been ongoing for months or years. This

requires you to readily make sense of the work that a team has already undertaken and the tools they have built so that you can effectively contribute to the team's efforts.

2. Work in the **industry** is much more collaborative/team-based. In academia, you own a research project and maybe pull in people for help when their expertise in a specific method is required. If no one in the lab has the required knowledge, you have to figure out how and where to learn the necessary skills. In industry, you own a small piece of the research project and are in constant communication with people working on other parts of the team. This allows for a faster pace of research. Additionally, a company has people with a wide range of skills. That means that you have access to guidance on topics outside your area of expertise and can thus pick up knowledge and skills more quickly or bypass the need to spend time learning altogether.

Do I need a degree to get a job in computational neuroscience? Are there many jobs?

Cesar:

Computational neuroscience is a vastly inter-disciplinary field. You don't need a degree in computational neuroscience to get a job doing computational neuroscience. If you have a strong mathematical or engineering background and some knowledge of biology, you are an attractive candidate. If you are currently a student, I would strongly recommend finding and pursuing an internship in the industry, if that's something that interests you. This will definitely give you a leading edge as an applicant, though it's not a necessity (I didn't do one during my time in school).

There are relatively few jobs doing computational neuroscience. However, there are two reasons why a background in computational neuroscience positively impacts your job prospects:

1. The field of neurotechnology is growing and computational neuroscience is the backbone of the industry so the number and variety of jobs available will also increase.
2. The skills you develop in computational neuroscience **are relevant to the analysis and interpretation of other physiological and sensor data**. The wearables industry is at a relatively more mature

stage and the jobs available in that industry tackle very interesting high-impact problems. For example, Verily recently released a [study](#) where they show that they can track the progress of Parkinson's disease using wearable data.

You can give only a single piece of advice to someone newly interested in computational neuroscience. What is your number one advice?

Artem:

Don't just sit and passively learn about computational neuroscience. Actually DO computational neuroscience. Our brains subconsciously fear the hard work. It is very tempting to just cruise through textbooks and papers, absorbing every bit of the knowledge. You feel like you're making some progress. *But this progress is mostly illusory.*

Without direct practice, it's impossible to advance. Go get your hands dirty. Go spend hours figuring out how to do something on your own, resisting the temptation to just look at the solution / copy-paste someone's code. It's better to come up with one clumsy inefficient script, written on your own, than to read tens of papers on a similar topic.

Cesar:

Stay curious. Don't be afraid to delve into a new topic. If there's something that piques your interest, take the time to learn it.

Authored and designed by [David Smehlik](#), edited by [Chelsea Lord](#), [Lina Cortéz](#), and [Sophie Valentine](#).

David has worked in (health) tech as a product designer with a background in molecular biology. Now he's looking to transition to neuroscience or neurotech in one way or another.

Chelsea works in clinical operations and is currently working for a company that is a leader in brain-computer interfaces. She studied neuroscience and is interested in the AI and Machine-Learning side of neurotechnology.

Lina is an electronics engineer and neurotechnology enthusiast who is highly interested in the application of brain-computer

interface in robotics control.

Sophie is a self-starter with a passion for using technology to benefit those who need it most, particularly in neurotechnology and mental health.