Tutorial: 3-D Cochlear Nucleus Frequency Model

A guide to the model described in:
A 3-Dimensional Model of Frequency Representation in the Cochlear Nucleus of the CBA/J Mouse
Muniak et al. (2012)
Journal of Comparative Neurology
DOI: 10.1002/cne.23238

http://3D.RyugoLab.com
Ver. 1 10-Oct-12

Our model is a quantitative sensory map that allows the user to make predictions about frequency representation at any 3-D coordinate location within the cochlear nucleus (CN) of the mouse. Further information about the development and discussion of this model can be found in the companion manuscript.

The model makes use of radial basis functions to describe tonotopy within the mouse CN. This analytical approach cannot be summarized by a succinct equation, but is instead governed by a large set of functions derived from our anatomical dataset. The model website was developed to allow other research groups to evaluate our functions.

This document is a tutorial on how to use this model, broken down into the following sections:

• Requirements
• Normalized coordinate systems
  o Registration to our CN template
  o Other mouse brain atlases
• Using the model software
• Interacting with the model
To use the model, there are three requirements:

1) **A set of 3-D coordinates**

   ![Table of 3-D coordinates]

   These coordinates represent locations within the CN at which you want to predict frequency representation. They could be based on the locations of labeled cell bodies, recording sites, or simply chosen arbitrarily. In order to make meaningful predictions, input coordinates must be based upon pre-defined coordinate systems (see below).

2) **A choice of subdivision**

   - **Choose Subdivision for Prediction:**
     - ![VCN]
     - ![DCN]

   The model is actually comprised of *two* predictive maps, one for the dorsal cochlear nucleus (DCN), and one for the ventral cochlear nucleus (VCN). To evaluate your coordinate data, you must specify which subdivision your data belong to. Of course, you can test your data against both subdivisions, but in all likelihood, only one will give meaningful predictions.

3) **The Silverlight plug-in**

   For ease of access, the model software has been compiled as a Silverlight plug-in, which is freely available and compatible with both Windows and Macintosh operating systems. The predictive functions are evaluated using your computer processor. Therefore, once loaded, the speed of evaluation is dependent upon your machine. To install Silverlight, go to [http://www.microsoft.com/getsilverlight/](http://www.microsoft.com/getsilverlight/).
NORMALIZED COORDINATE SYSTEMS

While our model can be evaluated at any coordinate in 3-dimensional Euclidian space, frequency predictions are obviously more reliable at locations proximal to the normalized location of the source data for the functions. To use our model, your CN data needs to be registered to the same reference frame as our template nucleus. Alternatively, your data can be defined in the coordinate system of a common mouse brain atlas, provided we have worked out the appropriate mapping of this atlas to our template. The following sections explain these concepts in further detail.

*Frequency model shown within Franklin & Paxinos (2008) mouse brain atlas.*
REGISTRATION TO OUR
COCHLEAR NUCLEUS TEMPLATE

Registering your data to our coordinate system is the most direct way to make predictions with the model. The absolute values of our coordinate system are arbitrary and do not directly reference particular stereotaxic landmarks. As such, your CN data can be sectioned in any manner you choose. A 3-D reconstruction of your data is used to register to our coordinate system. This section demonstrates one method for achieving this goal using the software package, Amira (http://www.amira.com/). This is by no means the only way to perform registration, but it is the method most commonly used by our lab. The basic workflow is as follows:

1) Collect serial-section image data of the CN

Ensure that your tissue is photographed in such a way that you can discern the borders of the CN. For best results, we recommend collecting as many sections as possible, capturing the entire nucleus. Obviously, keep track of your magnification factor... always have a scale bar handy!

2) Register serial-section data along the orthogonal axis

For example, if your sections were collected in the coronal plane, then your images need to be aligned to establish continuity along the anterior-posterior axis. This can be easily done using layers in Photoshop. Other software packages (e.g., Neurolucida, Amira) also have mechanisms for image registration. Blood vessels and other anatomical landmarks can serve as guides for alignment. Only rotational and translational manipulations should be applied to each section. Non-destructive adjustments (e.g., levels) can be applied to enhance visualization of regions of interest. Once all sections are aligned, export the dataset as an image-stack (i.e.,
consecutively numbered image files). If multiple imaging modalities are used (e.g., brightfield and fluorescence), make sure that all images from the same section are kept in alignment, and that the same bounding-box or crop-window is used for images from each modality. This is crucial for ensuring all data originating from the same experimental case are subject to the same coordinate manipulations (see below).

3) **Import image-stack data into Amira.**
   We prefer to do this by creating a .txt file in the same directory as the image-stack files. This file specifies the voxel-size and placement of individual images along the z-axis. The advantage of this format is that it allows you to import image-stacks that may be irregularly spaced (e.g., missing sections). For further information, see the Amira documentation.

   A typical file may look like this:

   ```
   # Amira Stacked Slices
   # (your notes here)

   # Pixel size in x- and y-direction (microns/pixel)
   pixelsize 1.845 1.845

   # Image list with z-positions
   mouse30_slice01.tif 0
   mouse30_slice02.tif 50
   mouse30_slice04.tif 150
   ...
   mouse30_slice12.tif 550
   end
   ```

   In this example the image resolution is 1.845 µm per pixel, the sections were cut at a thickness of 50 µm, and the 3rd section was lost. Opening this .txt file in Amira will import the image-stack. (Note: images must be listed in ascending order along the z-axis.) Attach an Orthoslice module to your image-stack to view the images in 3-D to make sure everything appears spatially correct.
4) **Segment the CN**

To create a 3-D surface of the CN, you first need to tell *Amira* where its borders are in your image-stack. To do this, attach a *LabelField* module, which will take you to the *Segmentation Editor*. (Note: *Amira* requires a single-channel, i.e. grayscale, image for segmentation. If your image-stack is in color, convert it by using the *ChannelWorks* module.) Demarcate the borders of the CN in each section in the *Segmentation Editor*. The lasso tool is useful for freehand drawing. A graphics tablet (e.g., *Wacom* products) is particularly helpful for this task.

5) **Resample the label field**

Now that you have an outline of the CN, you can resample the labels to reduce the computational burden. Attach a *Resample* module to the label field. We typically resample the x- and y-axes by 4X, giving a voxel size ~7-8 µm along these two dimensions. *We do not recommend resampling orthogonal to the cutting plane*, as your data is probably already sparse enough along this dimension.
6) **Generate a 3-D surface of the CN**

Attach a *SurfaceGen* module to your newly resampled label field. If your image-stack spans the entire nucleus, leave "add border" checked. If you are missing one, or both, end(s) of the nucleus, leave this option unchecked. Check "compactify", and uncheck any other options. For smoothing, you can choose "none" for the most literal representation of your label data, or "unconstrained smoothing" for a more polished approximation. We frequently apply smoothing, but the results depend on the dataset.

7) **Close any surface holes**

If you had left "add border" unchecked in the last step, your 3-D surface may appear to have a big hole on one side. To close this hole, go to the *Surface Editor*, and in the menu bar choose *Surface > Edit > Fill Hole*. This method is used for incomplete nuclei because the "cut" surface will now be represented by only one vertex point. This will minimize the contribution of this "cut" surface to the subsequent alignment algorithm.

8) **Simplify the 3-D surface**

The efficiency of the alignment algorithm depends on how many points are used to represent the 3-D surface. Chances are, the output from *SurfaceGen* will still contain a large number of vertices, oversampling the surface. With our data, we typically reduce the number of vertices by ~20% so that we are representing a CN surface with ~7000-9000 points. This can be achieved two different ways: using the *RemeshSurface* module, or the *Simplifier Editor.*
If you have the Mesh option installed with Amira, use the RemeshSurface module. Choose "best isotropic vertex placement", adjust the "Desired size %" to get the appropriate number of vertices, and interpolate "smoothly".

Without the Mesh option installed, you can still accomplish the goal of downsampling, though the resultant vertices will not be evenly distributed. Open the Simplifier Editor for the surface. At top will be the current number of points and faces. Below, enter a new number of faces (you'll have to do the % calculation yourself) and click "Simplify now".

9) Register 3-D surface to our template
First, you'll need to load our template CN. If you haven't already downloaded this file, go to the Downloads section of the website to get it. Once loaded, compare the template to your data... are they in the same orientation? In other words, are the principal axes roughly the same, or is one the mirror image of the other?

If your data does not match, use the Transform Editor to make the appropriate gross adjustments (flip surface, rotate 90 deg., etc). Note: DO NOT apply the transform to your surface! If you apply, the transformation matrix we are working towards will no longer map your original image dataset to the model, but will instead map from the coordinates of this newly modified surface.
Once your surface is roughly in the same orientation as our template, we can begin automated alignment. Connect the \textit{AlignSurfaces} module to your simplified CN surface. Make sure your CN surface is designated as the "\textit{Surface_to_be_transformed}". Next, connect the module to the template CN, designating it as the "\textit{Reference_surface}". To get your surface in the right ballpark, click the button that says "\textit{Centers}".

Now, set up the parameters for automated alignment: Leave "iterate" checked, "use correspondence" unchecked. For "\textit{Trafo}" (i.e., transformation), we tend to use the affine transformation, as explained in our manuscript. Considering there is a good chance that independent histological procedures may result in differential shrinkage of your brain tissue, you should at least consider this option. We max out the "\textit{Stop}" parameters: "relative RMS" is set to 1e-12 (admittedly overkill), and "max iter" is set to 1000. Click the "\textit{Surfaces}" button to begin alignment! If you have attached viewers to each of the surfaces, you should be able to watch the action in real time.
10) Get your transformation

Once your CN surface has been satisfactorily registered to the template CN, you’ll need to extract the final transformation matrix. Go to the Amira command prompt, called the Console, and type:

[your surface name] getTransform

and press enter. You should get a string of 16 numbers. Save this string in a safe place.

11) Map your data to our template

Now that you have a transformation matrix, you are ready to map all of your coordinate data from your experimental CN to the coordinate system of our model. This requires some basic matrix multiplication. Suppose your transformation string is made up of the values:

\[
\begin{bmatrix}
T_1 & T_2 & T_3 & \ldots & T_{16}
\end{bmatrix}
\]

Reformat these values into the 4x4 matrix:

\[
\begin{bmatrix}
T_1 & T_2 & T_3 & T_4 \\
T_5 & T_6 & T_7 & T_8 \\
T_9 & T_{10} & T_{11} & T_{12} \\
T_{13} & T_{14} & T_{15} & T_{16}
\end{bmatrix}
\]

This is the equivalent of the Matlab function (note the transpose at the end):

\[
\text{reshape([T1...T16],4,4)'}
\]

Next, format your 3-D coordinate data into columns, adding a 4th column of ones, to get:

\[
\begin{bmatrix}
X_1 & Y_1 & Z_1 & 1 \\
X_2 & Y_2 & Z_2 & 1 \\
X_3 & Y_3 & Z_3 & 1 \\
\ldots \\
X_n & Y_n & Z_n & 1
\end{bmatrix}
\]

To map your data to our coordinate system, simply perform the matrix multiplication:

\[
\text{MAPPED\_DATA} = \text{DATA\_MATRIX} \times \text{TRANSFORM\_MATRIX}
\]

Drop the 4th column of the result, and you have your mapped coordinates! You are now ready to make predictions with our model (see below).
OTHER MOUSE BRAIN ATLASES

If you are unable, or simply do not wish, to register your data to our template CN, we also support the ability to make predictions with our model using the coordinate system of a standardized brain atlas. However, a word of caution: using this approach, while potentially simplifying your procedure, will undoubtedly introduce additional uncertainty in the accuracy of predictions. This is because two approximations are used in serial order. First, you are approximating your data to the reference frame of a brain atlas. Next, you are using our approximation of the brain atlas to the coordinate system of our model. We have not rigorously tested these mappings, and offer them for use at your own discretion.

At present, we have worked out a mapping for the mouse brain atlas of Franklin & Paxinos (2008) to our template CN. To calculate this mapping, we simply created 3-D surfaces of the CN using the atlas plates, with the stereotaxic Bregma-centered coordinate system as the reference frame. Each CN surface was then aligned to our template CN using the procedures described above.

Note: We observed that each sectioning plane shown in the atlas (coronal, horizontal, sagittal) places the CN at a slightly different location (left). To address this variability, we computed independent mappings for each plane. One should also be aware that this brain atlas is based on the C57Bl/6J mouse strain, whereas our model was constructed using data from CBA/J mice. Also keep in mind that if you use interaural coordinates, you must convert them to Bregma-centered coordinates first!
USING THE MODEL SOFTWARE

Making predictions with the online model software is quite straightforward. The interface is divided into two sections: Input settings are configured on the left, and results are displayed at right.

Choose Subdivision: You must indicate which prediction model you wish to use, DCN or VCN.

Output Decimal Points: Adjust specificity of prediction output.

Output Format: Predictions can be made with respect to frequency (in kHz) or cochleotopic position (% distance from apex). The two values are interchangeable based on our place-frequency map of the CBA/J mouse cochlea: 

\[
\%_{\text{dist. from apex}} = 78.43 \times \log_{10}(\text{freq.}) - 49.96
\]

Coordinate System: Specify which reference frame your data is based in (see previous section for details). If your data is mapped to our template CN, choose Muniak et al. (2012), which is the default option. If your data is based upon a mouse brain atlas, choose the appropriate setting (if available).

Input Coordinates: Enter one set of 3-D coordinates per line. Numbers can be separated by any combination of spaces, tabs, commas, or semi-colons. You can easily copy & paste data from Excel, Matlab, etc.

Compute Prediction: Hit the button to begin computation. A status indicator will show progress. Results are shown in the Model Prediction box, with one prediction made per input 3-D coordinate.
INTERACTING WITH THE MODEL

We have created a reference guide for exploring tonotopy in the mouse CN. This model consists of a series of "virtual slices" made along various sectioning planes (similar to Figure 10 in the manuscript). Where possible, we have also attempted to include stereotaxic coordinates of standard brain atlases to aid experimental approaches. This reference guide can be found in the atlas section of the website.

In addition, users may wish to directly explore the frequency model in a 3-D environment. To facilitate this, we have supplied a tetrahedral prediction dataset in the downloads section. This is the same file used to create Figures 9 and 10 in the manuscript. This file can be opened with various software packages, both commercial (e.g., Amira) and open-source (e.g., Paraview). Below, we demonstrate how you can explore this model in Paraview, including creating your own "virtual slices".
1) **Download the dataset**
   Included in the main package are 4 files:

   **CNmodel_DATA.inp**
   This is the predicted dataset. The file is an AVS-compatible UCD structure that can be read by multiple programs. Contained are the coordinates of tetrahedral grids of the DCN and VCN, as well as predicted cochleotopic values at each node. Coordinates are in the reference frame of our template CN. *Note: By default, predictions are supplied as cochleotopic values* (see paper for discussion). To convert to frequency values, simply use our place-frequency map of the CBA/J mouse cochlea.

   **CNmodel_DEMO.pvsm**
   This is a readymade demonstration file that can be loaded in Paraview for viewing the model (see next step).

   **colorJet_Paraview.xml**
   Color map file formatted for use in Paraview. It is the equivalent of the "jet" color map from Matlab applied to the interval [0,100].

   **colorJet_Amira.icol**
   Color map file formatted for use in Amira.

2) **Install Paraview**
   If you haven't already installed Paraview on your machine, head to [http://www.paraview.org/](http://www.paraview.org/) and download the latest version. Binary installers are available for all common operating systems.

3) **Load the demo file**
   In Paraview, go to *File -> Load State*, and choose "CNmodel_DEMO.pvsm". This will set up a basic viewing platform for the 3-D model.

   *Note: Currently, upon opening the demo file, Paraview will require you to locate the dataset file, "CNmodel_DATA.inp", even though it is in the same directory. You must manually locate the file or else you will receive an error! Hopefully this will be fixed in future versions of Paraview. Otherwise, everything else is automatic, and you will be presented with a demonstration of the frequency model.*
4) **Explore the demo model**

In *Paraview*, you can freely rotate the model in any direction. 4 special modules have been applied to the model: 3 clipping planes are set up in parasagittal, horizontal, and paracoronal views. A contour viewer is also set up showing iso-cochleotopic (a.k.a., iso-frequency) contours at 20% intervals from 0% - 100%.

To adjust a clipping plane, click on the appropriate module in the *Pipeline Browser*. The plane will be indicated with a red outline. Click anywhere on this plane and drag the cursor to slide the clipping plane along the orthogonal axis. To modify the direction of the clipping plane, click and drag the white bar representing the orthogonal axis. To create the new clipping plane, you must click the "Apply" button in the *Properties* tab. Optionally, you can have *Paraview* instantly render any changes by turning on the "Apply changes to parameters automatically" button in the toolbar.

To modify or create new contours, click the contour module in the *Pipeline Browser*, and go to the *Value Range* box in the *Properties* tab. Use the buttons to the right to add/delete values. **Remember that the model values are on a cochleotopic scale, not frequency.**
5) **Create a virtual slice**

Turn off all special modules by clicking the "eye" next to each one in the **Pipeline Browser**. Click on the main model, "CNmodel_DATA.inp", in the **Pipeline Browser**, and turn on its visibility by clicking the eye. In the toolbar, select the "Slice" icon. This will create a slice module that operates just like the clipping plane module. Drag the slice around to change its position. In the **Properties** tab, you can orient the slice along one of the primary axes. Don't forget to "Apply".

For some fun, you can also create multiple slices simultaneously. In the **Value Range** box, click "New Range", and enter the following:

- **From**: -1500
- **To**: 1500
- **Steps**: 31

and click OK. This will create multiple parallel slices spaced at 100 µm intervals, centered on the slicing plane.
We have also created additional *Paraview* state files that allow you to explore the model with respect to stereotaxic coordinates. These are essentially interactive stereotaxic frequency atlases of the CN. See the downloads section of the website for these files.

If something was not explained clearly, or you have any other questions, comments, etc., please do not hesitate to contact us!

**Michael Muniak, PhD**

*m.muniak [at] garvan.org.au*

http://www.ryugolab.com/