SEEG Electrode Trajectory Analysis Module: Comprehensive Documentation

Rocío Ávalos

June 3, 2025

Contents

1 Introduction

The SEEG (Stereotactic Electroencephalography) Electrode Trajectory Analysis Module is a comprehensive Python framework designed for analyzing electrode trajectories in 3D brain space. This module provides functionality for clustering, community detection, trajectory analysis, and quality assessment of SEEG electrode placements using advanced machine learning and geometric analysis techniques.

1.1 Primary Objectives

- Automated detection and clustering of electrode contact points
- Trajectory reconstruction from bolt heads to brain entry points
- Quality assessment and validation of electrode placements
- Hemisphere-specific analysis capabilities
- Interactive visualization and reporting

2 System Architecture and Dependencies

2.1 Core Dependencies

The module requires the following Python packages:

- **3D Slicer**: Medical image processing platform
- **Scientific Computing**: NumPy, SciPy, scikit-learn
- **Machine Learning**: DBSCAN clustering, PCA, NetworkX (Louvain)
- **Visualization**: Matplotlib, Plotly (optional)
- **Data Processing**: Pandas, scikit-image

2.2 Input Data Requirements

1. **Electrode Mask Volume**: Binary volume containing electrode contact locations
2. **Brain Mask Volume**: Segmented brain tissue for surface normal calculations
3. **Combined Volume** (optional): Integrated bolt heads, entry points, and trajectory paths
4. **Entry Points Volume** (optional): Brain surface entry point locations
5. **Bolt Heads Volume** (optional): Cranial bolt head positions

3 Core Analysis Methods

3.1 Coordinate Extraction and Processing

The analysis begins with extracting electrode coordinates from medical imaging volumes:

```
def get_all_centroids(volume):
    """Extract centroids from volume using connected components
       """
    array = get_array_from_volume(volume)
    labeled = label(array, connectivity=3)
    props = regionprops_table(labeled, properties=['centroid'])
    return convert_to_ras_coordinates(props)
```

3.2 Hemisphere Filtering

The module supports hemisphere-specific analysis using RAS coordinate system:

$$\text{Hemisphere} = \begin{cases} \text{Left} & \text{if } x_{RAS} < 0 \\ \text{Right} & \text{if } x_{RAS} > 0 \\ \text{Both} & \text{no filtering} \end{cases} \tag{1}$$

3.3 Trajectory Detection Methods

3.3.1 Method 1: Clustering-Based Approach

DBSCAN Clustering Uses density-based spatial clustering to group electrode contacts:

$$\text{DBSCAN}(\epsilon, \text{MinPts}) = \{C_1, C_2, \ldots, C_k, N\} \tag{2}$$

$$\text{where } C_i = \text{cluster } i,\ N = \text{noise points} \tag{3}$$

Parameters:

- ϵ: Maximum neighbor distance (default: 7.5mm)
- MinPts: Minimum neighbors for core points (default: 3)

Adaptive Parameter Selection When enabled, the algorithm automatically optimizes clustering parameters:

iteration $i = 1$ to max_iterations Apply DBSCAN with current parameters Calculate quality score based on expected contact counts Adjust parameters based on clustering results score improves Update best parameters

Louvain Community Detection Complementary analysis using graph-based community detection:

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j) \tag{4}$$

where Q is modularity, A_{ij} is adjacency matrix, k_i is node degree, and $\delta(c_i, c_j)$ indicates same community membership.

3.3.2 Method 2: Combined Volume Approach

Direct trajectory extraction from pre-segmented volume containing:

- Bolt heads (value = 1)
- Entry points (value = 2)
- Trajectory lines (value = 3)

3.4 Principal Component Analysis (PCA)

For each identified cluster, PCA determines the principal trajectory direction:

$$\mathbf{C} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T \tag{5}$$

$$\mathbf{v}_1 = \text{eigenvector with largest eigenvalue of } \mathbf{C} \tag{6}$$

Linearity score: $L = \frac{\lambda_1}{\lambda_1+\lambda_2+\lambda_3}$

4 Quality Assessment Framework

4.1 Contact Count Validation

Validates clusters against expected electrode configurations:

- Standard configurations: [5, 8, 10, 12, 15, 18] contacts
- Tolerance: ± 2 contacts for "close" matches
- Exact matches preferred for validation

4.2 Spacing Validation

Analyzes inter-contact spacing along trajectories:

$$d_i = \|\mathbf{p}_{i+1} - \mathbf{p}_i\|, \quad i = 1, \ldots, n-1 \tag{7}$$

Expected spacing range: 3.0-5.0mm (configurable)
Validation metrics:

- Mean spacing: $\bar{d} = \frac{1}{n-1}\sum_{i=1}^{n-1} d_i$
- Coefficient of variation: $CV = \frac{\sigma_d}{\bar{d}}$
- Valid percentage: $\frac{\text{valid spacings}}{\text{total spacings}} \times 100\%$

4.3 Contact Angle Analysis

Measures trajectory curvature between consecutive contact segments:

$$\mathbf{v}_1 = \mathbf{p}_i - \mathbf{p}_{i-1} \tag{8}$$

$$\mathbf{v}_2 = \mathbf{p}_{i+1} - \mathbf{p}_i \tag{9}$$

$$\theta_i = \arccos\left(\frac{\mathbf{v}_1 \cdot \mathbf{v}_2}{\|\mathbf{v}_1\|\|\mathbf{v}_2\|}\right) \tag{10}$$

Curvature angle: $\alpha_i = \theta_i$ (0° = straight, 180° = complete reversal)
Threshold for flagging: typically 40° (configurable)

4.4 Entry Angle Validation

Validates trajectory entry angles against surgical constraints:

$$\text{Entry Angle} = 90 - \arccos(|\mathbf{d} \cdot \mathbf{n}|) \tag{11}$$

where $\mathbf{d}$ is trajectory direction and $\mathbf{n}$ is surface normal.
Ideal range: 30°-60° with skull surface.

5 Trajectory Refinement

5.1 Hemisphere Splitting

For trajectories crossing the midline (x = 0), automatic splitting creates separate hemispheric segments.

5.2 Merging and Splitting Operations

Merging Criteria:

- Endpoint distance ¡ 15mm
- Direction angle difference ¡ 20°
- Combined count approaches expected electrode size

Splitting Criteria:

- Contact count ¿ maximum expected (default: 20)
- Multiple expected configurations can sum to actual count
- Spatial separation detectable by sub-clustering

6 Algorithmic Quality Scoring

6.1 Comprehensive Scoring Function

Each trajectory receives a quality score (0-100) based on multiple criteria:

$$\text{Score} = w_1 S_{\text{count}} + w_2 S_{\text{linearity}} + w_3 S_{\text{spacing}} \tag{12}$$
$$+ w_4 S_{\text{angles}} + w_5 S_{\text{length}} + w_6 S_{\text{entry}} \tag{13}$$

Default weights: $w_1 = 0.25, w_2 = 0.20, w_3 = 0.20, w_4 = 0.15, w_5 = 0.10, w_6 = 0.10$

6.1.1 Individual Score Components:

Contact Count Score (S_{count}):

$$S_{\text{count}} = \begin{cases} 25 & \text{if exact match with expected} \\ 20 - 2.5 \times |d| & \text{if } |d| \leq 2 \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

Linearity Score ($S_{\text{linearity}}$):

$$S_{\text{linearity}} = \begin{cases} 20 & \text{if } L \geq 0.95 \\ 10 + 10 \times \frac{L - 0.85}{0.10} & \text{if } 0.85 \leq L < 0.95 \\ 10 \times \max(0, \frac{L - 0.70}{0.15}) & \text{otherwise} \end{cases} \tag{15}$$

7 Output Generation and Visualization

7.1 Report Types

7.1.1 PDF Reports

1. **Trajectory Analysis Summary**: Overview statistics and parameters
2. **3D Visualizations**: Interactive and static trajectory plots
3. **Validation Reports**: Contact count, spacing, and angle validation
4. **Quality Assessment**: Individual trajectory scoring and recommendations

7.1.2 Interactive Visualizations

- **Plotly 3D Plots**: Rotatable, zoomable trajectory visualization
- **HTML Reports**: Interactive annotation interface for manual validation
- **Color-coded Quality**: Visual quality indicators based on algorithmic scores

7.2 Data Exports

7.2.1 CSV Files

- **Trajectory Scores**: Complete feature set for machine learning
- **Flagged Segments**: Non-linear trajectory segments requiring review
- **Validation Results**: Pass/fail status for each quality metric

7.2.2 Volume Outputs

- **Trajectory Lines**: 3D volume with reconstructed trajectory paths
- **Cluster Labels**: Volume with cluster assignments for each voxel

8 Configuration Parameters

8.1 Analysis Parameters

Table 1: Key Configuration Parameters

Parameter	Default Value	Description
`hemisphere`	'both'	Hemisphere filtering ('left', 'right', 'both')
`max_neighbor_distance`	7.5mm	DBSCAN epsilon parameter
`min_neighbors`	3	DBSCAN minimum samples
`expected_contact_counts`	[5,8,10,12,15,18]	Valid electrode configurations
`expected_spacing_range`	(3.0, 5.0)mm	Valid inter-contact spacing
`angle_threshold`	40°	Maximum curvature before flagging
`duplicate_threshold`	0.5mm	Distance threshold for duplicate detection
`use_adaptive_clustering`	False	Enable automatic parameter optimization
`validate_entry_angles`	True	Perform surgical angle validation

8.2 Output Control Parameters

Table 2: Output Configuration Options

Parameter	Default Value	Description
`use_original_reports`	True	Generate comprehensive PDF reports
`create_plotly_visualization`	False	Generate interactive 3D plots
`create_interactive_annotation`	True	Generate HTML annotation interface
`detect_duplicates`	True	Perform duplicate centroid analysis
`validate_spacing`	True	Perform inter-contact spacing validation
`analyze_contact_angles`	True	Perform trajectory curvature analysis

9 Restrictions and Limitations

9.1 Technical Constraints

- **Minimum Data Requirements**: At least 3 contacts per trajectory for meaningful analysis
- **Memory Limitations**: Large datasets (¿10,000 contacts) may require parameter adjustment
- **Browser Storage**: Interactive visualizations cannot use localStorage in Claude.ai environment
- **Coordinate System**: Assumes RAS (Right-Anterior-Superior) coordinate convention

9.2 Algorithm Limitations

- **Clustering Sensitivity**: DBSCAN performance depends on appropriate epsilon selection
- **Trajectory Linearity**: Assumes approximately linear electrode trajectories
- **Surface Normal Estimation**: Requires sufficient brain surface detail for angle validation
- **Hemisphere Boundary**: Simple x=0 plane may not reflect true anatomical midline

9.3 Validation Assumptions

- **Standard Electrodes**: Validation against predefined contact counts [5,8,10,12,15,18]
- **Uniform Spacing**: Assumes approximately uniform 3-5mm inter-contact spacing
- **Surgical Constraints**: Entry angle validation assumes 30-60° optimal range
- **Image Quality**: Requires adequate resolution for accurate centroid extraction

10 Usage Examples

10.1 Basic Analysis

```
# Standard bilateral analysis
results = main(
    use_combined_volume=True,
    hemisphere='both',
    use_adaptive_clustering=False
)
```

10.2 Hemisphere-Specific Analysis

```
# Left hemisphere only with adaptive clustering
left_results = main(
    hemisphere='left',
    use_adaptive_clustering=True,
    max_iterations=10
)

# Compare hemispheres
comparison = compare_hemispheres()
```

10.3 Advanced Quality Assessment

```
# Comprehensive analysis with all validation options
results = main(
    use_combined_volume=True,
    use_adaptive_clustering=True,
    validate_spacing=True,
    validate_entry_angles=True,
    analyze_contact_angles=True,
    refine_trajectories=True,
    create_plotly_visualization=True
)
```

11 Future Development

11.1 Planned Enhancements

- **Machine Learning Integration**: Supervised learning from manual annotations
- **Real-time Processing**: Streaming analysis capabilities for surgical navigation
- **Multi-modal Integration**: Incorporation of DTI and fMRI data
- **Standardized Validation**: Integration with clinical outcome measures

11.2 Research Applications

- **Surgical Planning**: Optimization of electrode placement strategies
- **Quality Control**: Automated assessment of implantation accuracy
- **Clinical Research**: Large-scale analysis of SEEG procedures
- **Training**: Educational tool for neurosurgical residents

12 Conclusion

The SEEG Electrode Trajectory Analysis Module provides a comprehensive framework for automated analysis and quality assessment of stereotactic electrode placements. Through the combination of clustering algorithms, geometric analysis, and validation metrics, it enables both clinical quality control and research applications in epilepsy surgery.

The modular design allows for flexible configuration based on specific clinical needs, while the comprehensive output generation supports both automated processing and manual review workflows. The hemisphere-specific analysis capabilities and adaptive parameter selection make it suitable for a wide range of clinical scenarios and electrode configurations.

References

[1] Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (pp. 226-231).

[2] Blondel, V. D., Guillaume, J. L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), P10008.

[3] Cardinale, F., Cossu, M., Castana, L., Casaceli, G., Schiariti, M. P., Miserocchi, A., ... & Lo Russo, G. (2013). Stereoelectroencephalography: surgical methodology, safety, and stereotactic application accuracy in 500 procedures. *Neurosurgery*, 72(3), 353-366.

[4] Jolliffe, I. T. (2002). *Principal component analysis* (2nd ed.). Springer-Verlag.