SegError Manual

Contents

1 Computational Walkthrough

This section should provide a high-level description of the computational methods and assumptions made by this package.

1.1 General Definitions and Assumptions

The entire package presumes the existence of two segmentations, which we refer to as S and T. The segmentations are further assumed to be convertible to n-dimensional arrays with the same dimensions, and containing only non-negative integer values. Currently, the package will only read a .tif file for each segmentation (though can be easily extended to handle h5 files, etc.). Further, the resulting arrays are converted to unsigned 64-bit integers upon import.

Given these points, we can make several definitions.

$$S \coloneqq \text{a partition of the voxels into disjoint subsets } \{S_1, S_2, ..., S_m\}$$
$$T \coloneqq \text{a partition of the voxels into disjoint subsets } \{T_1, T_2, ..., T_l\}$$
$$(T \text{ is the ground truth segment if applicable})$$

$$N \coloneqq \#\text{ voxels in } S = \#\text{ voxels in } T$$

$$s_i \coloneqq |S_i| \qquad t_j \coloneqq |T_j| \qquad c_{ij} \coloneqq |S_i \cap T_j|$$

$$s'_i \coloneqq \frac{s_i}{N} \qquad t'_j \coloneqq \frac{t_j}{N} \qquad p_{ij} \coloneqq \frac{c_{ij}}{N}$$

As will be important for all of the metric formulations, note briefly that

$$s_i = \sum_j c_{ij} \qquad t_j = \sum_i c_{ij}$$
$$s'_i = \sum_j p_{ij} \qquad t'_j = \sum_i p_{ij}$$

Throughout this document, **score** metrics refer to those which increase as the segmentations become more similar, and **error** metrics refer to those which do the opposite.

1.2 High-Level Outline

There are currently two main phases of computation within SegError. The first preprocesses the segmentations and calculates the overlap matrix, while the second computes the desired metrics using the overlap matrix representation. We'll review each of these steps chronologically for the standard usage.

1.3 2D-Relabelling

If the user has specified that SegError should compute any 2-dimensional versions of a metric, then we first generate a version of the segmentation where segment ids are specific to an xy slice (aside from the '0' segment, which is preserved in this step across both S and T).

We perform this transformation through connected components analysis specific to a 2d slice. This amounts to multiple graph traversals over each 2d slice, where each traversal assigns a new id to voxels which both have the same segment id and are connected through some path of other voxels that also have the same segment id.

1.4 Foreground Restriction

By default, SegError will limit the calculation of any metrics below to voxels which have nonzero label within T. Users can disable this option using the -nofr flag.

This is directly handled by NumPy's indexing. First, we generate a view of S where $T! = 0$, perform the same for T, and then simply return these views (1d arrays in most cases).

1.5 Overlap Matrix Calculation

The last step of the 'preprocessing' steps is the actual computation of the overlap matrix. The overlap matrix finds the value of c_{ij} over all pairs between a segment of S and a segment of T.

This process contains a hidden subtlety for some processing pipelines. Specifically, if the predicted segmentation was generated by some versions of seung-lab watershed code, any singleton voxel would be unilaterally assigned to the '0' segment, regardless of its relation to the background of the overall image. In order to handle this case, we provide an option (on by default) to re-split the '0' segment into singleton voxels over the segmentation volume at this step. Users can skip this step by the '-no_split0' flag at the command-line.

After potentially dealing with the '0' segment, we then create a new sparse matrix over the segmentation ids using SciPy's utilities.

2 Rand Index-Based Metrics

Let,

$$TP = \#\text{ voxel pairs mapped to the } \textbf{same} \text{ segment within } \textbf{both } S \text{ and } T$$
$$FP = \#\text{ voxel pairs mapped to the } \textbf{same} \text{ segment within } S \text{ but not } T$$
$$FN = \#\text{ voxel pairs mapped to the } \textbf{same} \text{ segment of } T, \text{ but not } S$$
$$TN = \#\text{ voxel pairs mapped to the } \textbf{different} \text{ segments of } \textbf{both } S \text{ and } T$$

There are two general ways we can derive Rand Index-Based Metrics for segmentaiton. One involves only counting **distinct** pairs of voxels, and the other also includes counting a voxel by itself.

2.1 Counting Pairs of Distinct Voxels

Recall the number of pairs of distinct voxels within a set of size $n = \binom{n}{2}$. Following this reasoning, we can note that $TP = \sum_{i,j} \binom{c_{ij}}{2}$, $TP + FP = \sum_i \binom{s_i}{2}$, $TP + FN = \sum_j \binom{t_j}{2}$. We can see this for $TP + FN$ by noting that each pair within a segment t_j can either be grouped together by S or not. If they are grouped together, then this pair is a TP pair by the definition above, and if not, then it's a FN pair. Similar reasoning follows for the other two quantities.

Using the forms above, we can form the **Rand Index** (RI) by normalizing the number of true positives and negatives over the total number of pairs.

$$RI = \frac{TP + TN}{\binom{N}{2}} = RS \tag{1}$$

In practice, this value could be computed in relation to the **Rand Error** (RE), which we can relate by a subtraction from 1.

$$\begin{aligned} RE &= 1 - RS \\ &= \frac{FP + FN}{\binom{N}{2}} \\ &= \frac{(TP + FP) + (TP + FN) - 2TP}{\binom{N}{2}} \end{aligned}$$

$$RE = \frac{\sum_i \binom{s_i}{2} + \sum_j \binom{t_j}{2} - 2\sum_{i,j} \binom{c_{ij}}{2}}{\binom{N}{2}} \tag{2}$$

Throughout SegError, we commonly decompose different metrics into two versions, one referring to the severity of *splits* a segmentation S makes (relative to T), and the other does the same for *mergers*.

$$RE_{\text{split}} = \frac{FN}{\binom{N}{2}} = \frac{\sum_j \binom{t_j}{2} - \sum_{i,j} \binom{c_{ij}}{2}}{\binom{N}{2}} \text{ , } RE_{\text{merge}} = \frac{FP}{\binom{N}{2}} = \frac{\sum_i \binom{s_i}{2} - \sum_{i,j} \binom{c_{ij}}{2}}{\binom{N}{2}} \tag{3}$$

2.2 Including Pairs of Non-distinct Voxels

By now allowing a voxel to form a pair with itself, the number of pairs for a group of size $n = n^2$. Then, $TP = \sum_{i,j} c_{ij}^2$, $TP + FP = \sum_i s_i^2$, $TP + FN = \sum_j t_j^2$, and the respective versions of the equations above take the following form. Note that this takes a condensed alternate form if we instead use the values divided by N (i.e. s'_i, t'_j, and p_{ij}).

$$sRE = \frac{\sum_i s_i^2 + \sum_j t_j^2 - 2\sum_{i,j} c_{ij}^2}{N^2} = \sum_i s_i'^2 + \sum_j t_j'^2 - 2\sum_{i,j} p_{i,j}^2$$

$$sRE_{\text{split}} = \sum_j t_j'^2 - \sum_{i,j} p_{ij}^2 \qquad sRE_{\text{merge}} = \sum_i s_i'^2 - \sum_{i,j} p_{ij}^2$$

2.3 Rand F-Score

We can take the version of this metric involving non-distinct pairs, and normalize using either s_i or t_j in order to similarly capture the decomposition between *split* and *merge* scores. This underlies the official error metric for the ISBI2012 competition (`http://brainiac2.mit.edu/isbi_challenge/home`), and is returned by the **Rand F-Score** (RFS) option of SegError, along with its counterpart error (RFE).

$$RFS_{\text{split}} = \frac{\sum_{i,j} p_{ij}^2}{\sum_j t_j'^2} \qquad RFS_{\text{merge}} = \frac{\sum_{i,j} p_{ij}^2}{\sum_i s_i'^2} \tag{4}$$

$$RFS = \frac{\sum_{i,j} p_{ij}^2}{\alpha \sum_i s_i'^2 + (1-\alpha) \sum_j t_j'^2} \tag{5}$$

$$RFE = 1 - RFS \tag{6}$$

2.4 Metric Relations

It is important to note that the components of the Rand F-Score are equivalent to versions of the RE counterparts with different normalization (following the choice of whether or not to count non-distinct pairs). We show the derivation below for the *split* component, where the same results follow for the *merge* component.

$$\begin{aligned}
RE_{\text{split}} &= \frac{FN}{\binom{N}{2}} \\
&= \frac{\sum_j \binom{t_j}{2} - \sum_{i,j} \binom{c_{ij}}{2}}{\binom{N}{2}} \\
&= \frac{\sum_j (t_j^2 - t_j) - \sum_{i,j} (c_{ij}^2 - c_{ij})}{N(N-1)} \\
&= \frac{\sum_j (t_j^2) - N - \sum_{i,j} (c_{ij}^2) + N}{N(N-1)} \\
&= \frac{\sum_j t_j^2 - \sum_{i,j} c_{ij}^2}{N(N-1)} \\
&= \frac{\sum_j t_j^2 - \sum_{i,j} c_{ij}^2}{\sum_j t_j^2} \frac{\sum_j t_j^2}{N(N-1)} \\
&= \frac{\sum_j t_j'^2 - \sum_{i,j} p_{ij}^2}{\sum_j t_j'^2} \frac{\sum_j t_j^2}{N(N-1)} \\
&= (1 - \frac{\sum_{i,j} p_{ij}^2}{\sum_j t_j'^2}) \frac{\sum_j t_j^2}{N(N-1)} \\
&= (1 - RFS_{\text{split}}) \frac{\sum_j t_j^2}{N(N-1)}
\end{aligned}$$

As referred to above, a similar process follows for the *merge* component

$$RE_{\text{merge}} = (1 - RFS_{\text{merge}}) \frac{\sum_i s_i^2}{N(N-1)}$$

2.5 Precision and Recall

Also, note that RFS_{merge} and RFS_{split} are equivalent to precision and recall scores (defined below) if we choose to count non-distinct voxel pairs.

$$precision = \frac{TP}{TP + FP} = \frac{\sum_{i,j} p_{ij}^2}{\sum_i s_i'^2} \qquad recall = \frac{TP}{TP + FN} = \frac{\sum_{i,j} p_{ij}^2}{\sum_j t_j'^2}$$

If we instead chose to only count pairs of distinct voxels, they would take this form instead.

$$precision = \frac{TP}{TP+FP} = \frac{\sum_{i,j}\binom{c_{ij}}{2}}{\sum_i \binom{s_i}{2}} \qquad recall = \frac{TP}{TP+FN} = \frac{\sum_{i,j}\binom{c_{ij}}{2}}{\sum_j \binom{t_j}{2}}$$

These second forms may be offered in a future version, but are not currently available.

3 Information Theory Based Metrics

We also use distance metrics for segmentations based on information theory.

Specifically, given two segmentations S and T, let,

$$H(S) = -\sum_i s'_i \log s'_i \qquad H(T) = -\sum_j t'_j \log t'_j$$
$$H(S|T) = -\sum_{i,j} p_{ij} \log \frac{p_{ij}}{t'_j} \qquad H(T|S) = -\sum_{i,j} p_{ij} \log \frac{p_{ij}}{s'_i}$$
$$I(S,T) = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{s'_i t'_j}$$

$H(S)$ is known as the **Shannon Entropy** for a segmentation S, and $I(S,T)$ to be the **Mutual Information** between S and T. $H(S|T)$ is the **Conditional Entropy** of S given T, which represents the entropy leftover within S after observing a particular value of T.

3.1 Variation of Information

We then calculate the **Variation of Information** (VI or VIE) as

$$VI(S,T) = H(S) + H(T) - 2I(S,T) = H(S|T) + H(T|S)$$

where we use the second form (made explicit below) for our current computation

$$VI = -\sum_{i,j} p_{ij}(\log \frac{p_{ij}}{s'_i} + \log \frac{p_{ij}}{t'_j}) \tag{7}$$

$$VI_{\text{split}} = H(S|T) = -\sum_{i,j} p_{ij} \log \frac{p_{ij}}{t'_j} \qquad VI_{\text{merge}} = H(T|S) = -\sum_{i,j} p_{ij} \log \frac{p_{ij}}{s'_i} \tag{8}$$

This is an unnormalized metric, and so computing the score version (VIS) just returns the inverse.

$$VIS = -1 * VI \tag{9}$$

3.2 Variation F Score

However, we also form a normalized metric for each of the above, and derive a similar **Variation F-score** ($VIFS$) as with the Rand Error. Note that $H(S) = I(S,T) + H(S|T)$

$$VIFS_{\text{split}} = 1 - \frac{H(S|T)}{H(S)} = \frac{I(S,T)}{H(S)} \qquad VIFS_{\text{merge}} = 1 - \frac{H(T|S)}{H(T)} = \frac{I(S,T)}{H(T)} \tag{10}$$

$$VIFS = \frac{I(S,T)}{\alpha H(T) + (1-\alpha)H(S)} \tag{11}$$