EPIC RESEARCH DIAGNOSTICS	Algorithm Requirements

ORIGINATOR:

Name	Job Title	Signature	Date
Andrew Mason	Chief Technology Officer		3/5/12

APPROVALS:

Name	Job Title	Signature	Date
Nancy Rizzo	Founder and CEO	N Rizzo	3/5/12.
Mike Stowell	V.P of Operations	M. Stowell	3/5/12
Brian Tompson	Software Engineer		3/5/12
Andrea Miller	Quality Assurance Manager	Andrea Miller	3/5/12

1.0 Purpose

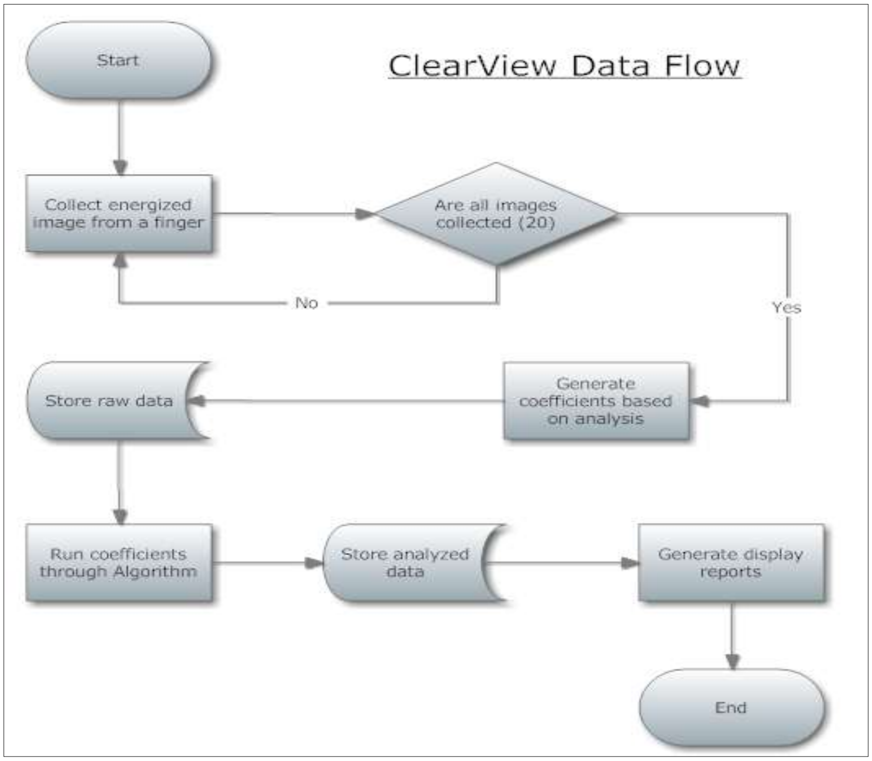
The purpose of this document is to describe the requirements for the process through which energized images are analyzed to produce a scaled score of 0-25 on a report. The same process is run for every sector on every finger. The process is run two times, once for the unfiltered energized images and once for the energized images.

2.0 Scope

The scope of this document consists of the algorithm processes as a whole, no other systems requirements are defined in this document.

3.0 Algorithm Requirements

The image below represents an ***overview*** of how the process flows, the process labeled "Run coefficients through Algorithm" is a container for a number of processes that will be described later in this document.



The algorithm can be further broken up into the following main sections that are processed in the order listed.

Raw Data Collection: This is the process of quantifying the images. The images are segmented into sectors that are then quantified by calculating 20 different coefficients for each sector. The same 20 coefficients are calculated for each and every sector.

EPIC Score Calculation: This is the process through which the EPIC score is calculated. The EPIC score is used in conjunction with the Logistic Regression score to determine the final output that is displayed on the report.

Logistic Regression Score Calculation: This is the process through which the Logistic Regression score is calculated. The Logistic Regression score is used in conjunction with the EPIC score to determine the final output that is displayed on the report.

Naive/Bayes Score Calculation: This is the process used to calculate the overall score for the report. This is calculated separate from the EPIC or LR scores and this value is not used anywhere else in the analysis process.

Report Score Selection: This is the process where the score to show on the report is selected between the EPIC score and the Logistic Regression score.

Report Production: This is the process of actually rendering the report on the screen for the user.

Each of the above sections will be further detailed below in order to fully describe the processes that are taking place. Some sections may contain tables, code fragments, illustrations, or references to other documents as a means to clarify the concepts.

Raw Data Collection

The generation of the report scores all begin with the analysis of a captured energized image. The images that are captured have the following properties:

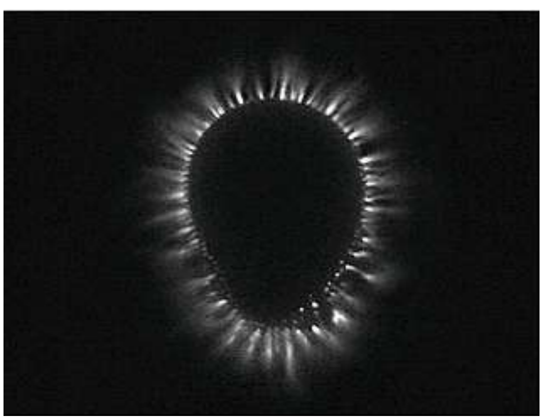
Size: 320 wide by 240 high (pixels)

Format: bmp

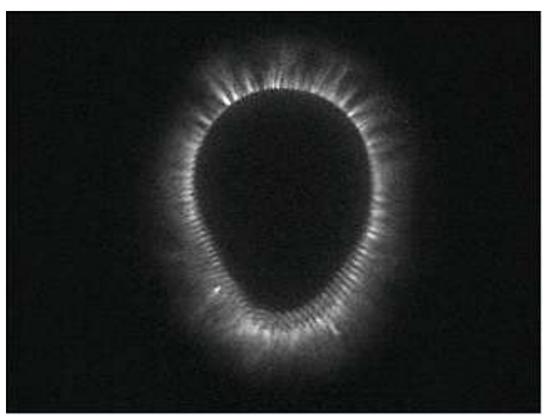
Grayscale

Pixel Intensity: 0 . . .255 (Dark . . .Light)

An image is captured for each finger on both the left and right hands, with the subject's finger resting directly on the glass electrode, this results in 10 images that are referred to hereafter as non-capacitive barrier images. A second set of 10 images (one for each finger) is taken, this time using a capacitive barrier (i.e., thin, electrostatic plastic in the size and shape of the glass electrode) between the finger and resulting in a set of images that will hereafter be referred to as capacitive barrier images. The images appear as show below:

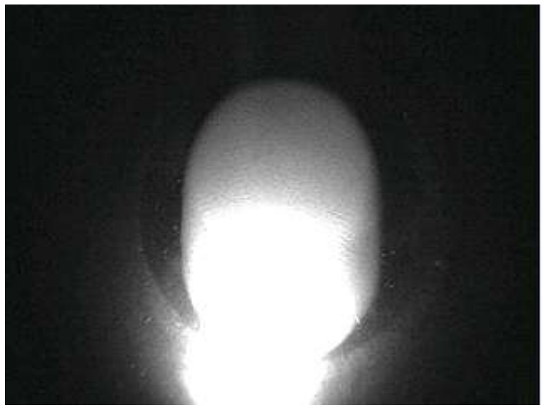


Non-capacitive Barrier Image



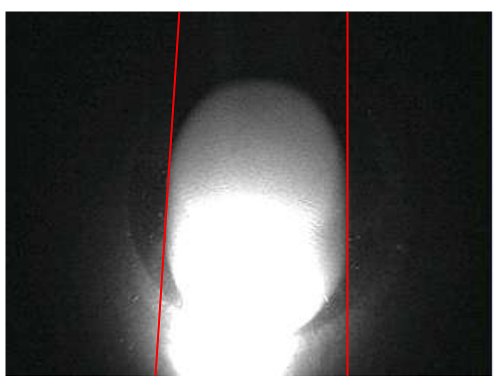
Capacitive Barrier Image

Just prior to the capture of the energized images show above, the system will capture an image hereafter referred to as a “finger image” or “lit finger image”. This image will be used in the next step discussed below, an example of a finger image is shown below:

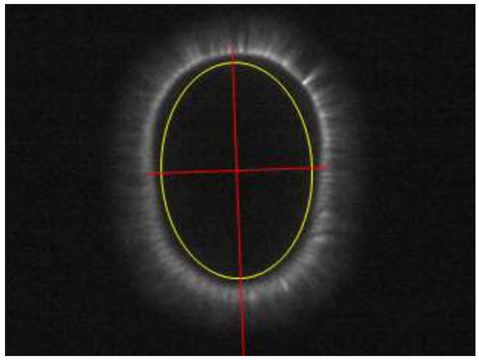


One the images have been captured, the alignment (i.e., angle of orientation and center-point coordinates) must be deciphered before the next step in the analysis process can take place, as the next step is dependent on the alignment value. The system will attempt to discern the alignment of the image using an algorithm that performs the following steps:

Analyze the finger image attempting to discern the angle the finger is at by recognizing the sides of the finger and calculating the angle of straight lines relative to 0 degrees . The two values are then averaged and the angle approximated. This approximate angle [A1] is the first factor in determining the finger orientation, and example of this is illustrated below:



The second component of the alignment calculation is to look at the energized image and determine the center point of the image and using a best fit oval algorithm to determine the oval that will fit into the captured image. Once this is done, the second factor used to determine the alignment value [A2] can be calculated, this is done by determining the angle of the vertical axis placed through the ellipse, an example of this process is shown below:



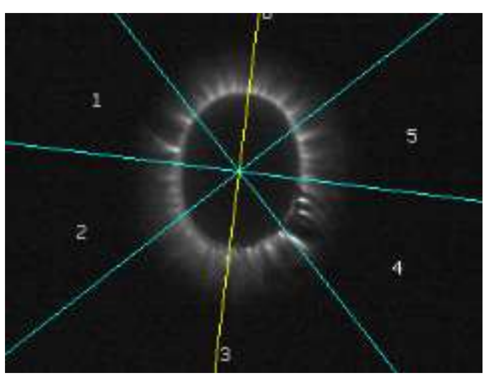
Once these two angles are determined, the following weighted equation is applied to the values to determine the calculated alignment value (angle). Most of the emphasis is placed on the value retrieved from the lit finger image:

$$Angle = (A1 * .9) + (A2 * .1)$$

This process of determining the center point and the angle is done for all 20 images and these alignment values will be used in the next step of the calculation.

The next step in the generation of the raw data is to segment image into individual sectors (or pie slices). Each pie slice will represent a section of the image from which quantified measurements will be calculated. A master list of the angle sizes that represent each sector for each finger is contained in another document entitled Finger Sector Map (SS-203-01).

The sectors are applied to the energized image in a counterclockwise pattern applying the angles relative to the alignment of the image. By way of example, if the image alignment was determined to be +5’ and the angle to apply was from 90’ to 120’, the actual angle applied would be from 95’ to 125’. An example for a finger image is shown below:



In the example above notice that the angles are applied relative to the alignment of the image, this is done for each energized image in the set.

Once the sectors have been defined, the process of calculating the coefficients for the sectors can be started. There are a total of twenty (20) different coefficients that are calculated for each of the sectors within an image, in the case of the image above there would be a total of 120 different coefficients calculated, 20 for each sector. The coefficients calculated are shown below in Table 1.

ClearView Scan Coefficient
Average Intensity
Average Intensity (1)
Average Intensity (2)
Average Intensity (3)
Average Intensity (4)
Break
Entropy
Fractal
Form
Form (1)
Form (2)
Form (3)
Form (4)
Form 2
Form 2 Prime
NS Integer
Normalized Area
Ring Intensity
Ring Thickness
Sector Area

A detailed description of each of the coefficients and the manner in which they are calculated is contained below:

Sector Area - The area is the number of pixels that have intensity greater than a specified noise level within a given sector. The pixels contained in the sector are determined by selecting all pixels inside the ellipse that are contained within a certain angle segment corresponding to this sector.

Normalized Area – This is the area value determined above, that is then normalized for the size of the sector that it represents. The normalization process is given by the formula:

Area Value * ((360 / #of Sectors)/ Angle of the sector)

Average Intensity - The Average Intensity per sector is computed by dividing the sum of intensities of all pixels in the sector area by the number of pixels in the sector area.

Average Intensity[1..4] – These are sub measurements of AI, using a ring segmentation process. The complete sector (pie slice) is broken down into 4 equal size segments and the AI measurement is made for each. AI1 is closest to the inner ring and AI4 is the furthest.

Entropy - The Entropy of each sector is computed by first computing the standard Shannon Entropy along each profile. The profiles are created in the following manner:

The image is traversed radially in a clockwise manner with the center point of the ellipse serving as midpoint. This is done in steps of 1/4 of an angular degree. For each of the resulting 360x4 = 1440 angles, an image profile is computed by choosing the pixels from the active area that intersect with a ray at one of the 1440 angles and centered at the ellipse midpoint.

The final value is determined by calculating the entropy of each profile within a sector and then averaging the result across all profiles within the sector.

Form - After having determined the active area of the finger image as described above, the image is traversed radially in a clockwise manner with the center point of the ellipse serving as midpoint. This is done in steps of 1/4 of an angular degree. For each of the resulting 360x4 = 1440 angles, an image profile is computed by choosing the pixels from the active area that intersect with a ray at one of the 1440 angles and centered at the ellipse midpoint. Furthermore, the active image area is divided into three concentric rings. The form value for each sector consists of three coefficients, one for each of the three rings. Each such coefficient is a weighted sum of three different coefficients C1,C2,C3.

The coefficient C1, the derivative coefficient, measures the amount of change in pixel intensity along each profile within a given ring. Amount of change is quantified by computing the derivative of the aforementioned profile.

The coefficient C2 measures whether there is a gap in the inner finger image contour and assigns a value corresponding to the size of such a gap.

The coefficient C3 measures whether there is an abrupt break (sudden drop in pixel intensity) along the profiles. For robustness against noise, several profiles are analyzed simultaneously for the presence of a break.

Form[1..4] - These are sub measurements of Form, using a ring segmentation process. The complete sector (pie slice) is broken down into 4 equal size segments and the AI measurement is made for each. Form1 is closest to the inner ring and Form4 is the furthest.

Form2 - The Form2 value is computed by using a weighted sum of the derivative coefficients for the first and second ring described in the Form calculation.

Form2Prime - The Form2Prime value is generally the same as Form2, but uses a different traversing method to obtain a measurement with a different perspective.

Break Coefficient - The break coefficient for each sector is computed by measuring whether there is an abrupt break (sudden drop in pixel intensity) along the profiles. For robustness against noise, several profiles are analyzed simultaneously for the presence of a break.

Fractal - The Fractal Dimension of the active area in a sector is computed by computing the fractal dimension (in its mathematical sense) using the standard box-counting method for a two-dimensional area.

NS Integer - The NS value is computed via the formula

$$\frac{\text{Average Intensity(Image sector) / Aver. Int.(Calibration sector)}}{\text{L\_Image / L\_Calibration}} - 0.5$$

Where:

L_Image = Log (Number of active pixels/Number of total pixels + epsilon) per each sector of the finger image.

L_Calibration = Log (Number of active pixels/Number of total pixels + epsilon) per each sector of the calibration image;

Here, the value epsilon = 10^(-4) is added for stability reasons to avoid that the values of L_image and L_Calibration become too small (which would happen in the rare case of numerator and denominator being almost identical).The value 0.5 in the above formula is subtracted for normalization purposes.

Ring Thickness - This is a process that measures the thickness radially in pixels of the bright portion of the inner ring. The value here can range from 0 to 20 or 30.

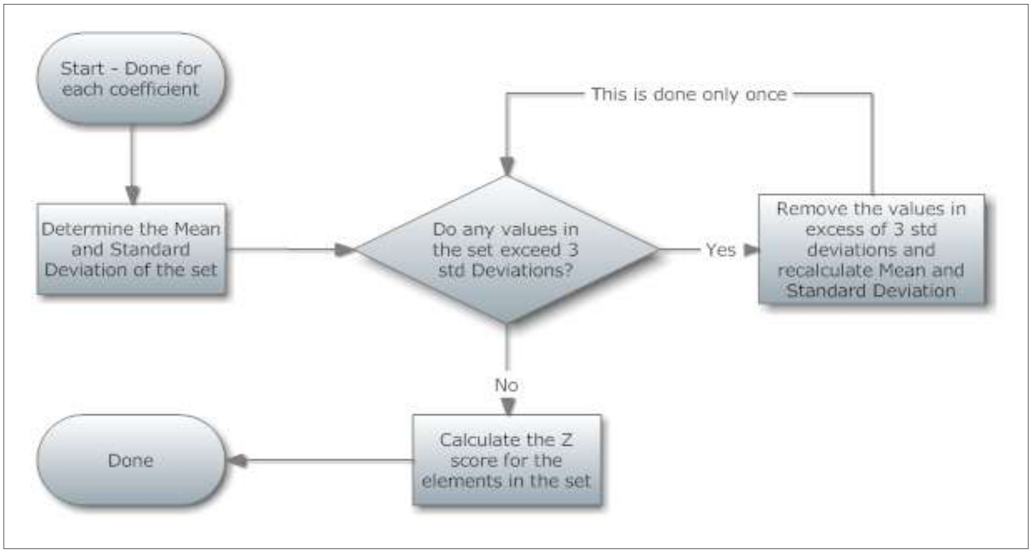
Ring Intensity – If the ring thickness coefficient registers, then the average intensity of the ring is calculated. This I accomplished by determining the intensity of all the pixels that make up the ring thickness portion of the calculation and averaging their intensity together.

When the values are calculated, they are truncated to four(4) decimal places before being stored in the database.

EPIC Score Calculation

The first of the score values to be calculated is the EPIC score, this is one of the base scoring components used to render the final report. An EPIC score is calculated for each of the sectors, this score is based on the coefficients calculated in the step described above.

The algorithm will perform the following processing on each of the listed coefficient data sets [Area, Average Intensity, Ai1, Ai2, Ai3, Entropy, Form, Form1-1, Form1-2, Form1-3, Form1-4, Form2, Form2Prime, Fractal, NS Integer]. The process will be independently completed for both the non-capacitive barrier images and the capacitive barrier images.



The Z-score is defined in statistics as:

The standard score is

$$z = \frac{x - \mu}{\sigma}$$

where:

- x is a raw score to be standardized;
- μ is the mean of the population;
- σ is the standard deviation of the population.

The Z-score ranking is then stored for each of the coefficients that were listed above, the three coefficients where Z-scores are not calculated are AI4, Ring Thickness and Ring Intensity. This is due to that fact that these values are often zero and are not a good candidate for this type of numerical analysis.

Once the Z-scores have been calculated for both capacitive barrier images as well as the non-capacitive barrier images, the following transform is applied to each of the coefficients on a per sector basis in order to get a base score for the sector. This is done independently for each set of images. Notice the transformation of the highlighted elements does not include Z-Score in the calculation.

Coefficient	Transformation
Average Intensity	Value * Z-score * 300.0
AI1	Value * Z-score * 25.0
AI2	Value * Z-score * 25.0
AI3	Value * Z-score * 25.0
Ai4	Value * 25.0
Break	N/A
Entropy	Value * Z-score * 1500.0
Fractal	Value * Z-score * 225.0
Form	Value * Z-score * 300.0
Form1-1	Value * Z-score * 75.0
Form1-2	Value * Z-score * 75.0
Form1-3	Value * Z-score * 75.0
Form1-4	Value * Z-score * 75.0
Form 2	Value * Z-score * 300.0
Form 2 Prime	Value * Z-score * 300.0
NS Integer	Value * Z-score * 3500.0
Normalized Area	Value * Z-score * .5
Ring Intensity	Value * 25.0
Ring Thickness	Value * 300.0
Sector Area	N/A

The EPIC Base score is then calculated using the following formulae, the values referenced are the values after the above transformations are applied. The following code is used to determine the significance score:

```
if (Math.Abs(Ai1Z) >= .9m || Math.Abs(Ai2Z) >= .9m || Math.Abs(Ai3Z) >= .9m)
```

```
{
   if (Math.Abs(NormalizedAreaZ) >= .9m && Math.Abs(Ai1Z) >= .9m)
   {
          SignificanceScore += 6000;
   }
   if (Math.Abs(NormalizedAreaZ) >= .9m && Math.Abs(Ai2Z) >= .9m)
   {
          SignificanceScore += 7000;
   }
   if (Math.Abs(NormalizedAreaZ) >= .9m && Math.Abs(Ai3Z) >= .9m)
   {
          SignificanceScore += 8000;
   }
}
else
{
   if (Math.Abs(NormalizedAreaZ) >= .9m && Math.Abs(AvgIntZ) >= .9m)
   {
          SignificanceScore += 5000;
   }

}

// Rule Set 2
// ----------
// [If Form1, Form2 and Entropy intensity hit above the .9 threshold, than add values in
these rules are cumulative]
Boolean Rule2Hit = false;
if (Math.Abs(Form1_1Z) >= .9m || Math.Abs(Form1_2Z) >= .9m || Math.Abs(Form1_3Z)
>= .9m || Math.Abs(Form1_4Z) >= .9m)
{
   if (Math.Abs(Form1_1Z) >= .9m && (Math.Abs(Form2Z) >= .9m ||
Math.Abs(Form2PrimeZ) >= .9m) && Math.Abs(EntropyZ) >= .9m)
   {
          SignificanceScore += 8000;
          Rule2Hit = true;
   }
   if (Math.Abs(Form1_2Z) >= .9m && (Math.Abs(Form2Z) >= .9m ||
Math.Abs(Form2PrimeZ) >= .9m) && Math.Abs(EntropyZ) >= .9m)
   {
          SignificanceScore += 9000;
          Rule2Hit = true;
```

```
  }
  if (Math.Abs(Form1_3Z) >= .9m && (Math.Abs(Form2Z) >= .9m ||
Math.Abs(Form2PrimeZ) >= .9m) && Math.Abs(EntropyZ) >= .9m)
  {
        SignificanceScore += 10000;
        Rule2Hit = true;
  }
  if (Math.Abs(Form1_4Z) >= .9m && (Math.Abs(Form2Z) >= .9m ||
Math.Abs(Form2PrimeZ) >= .9m) && Math.Abs(EntropyZ) >= .9m)
  {
        SignificanceScore += 11000;
        Rule2Hit = true;
  }
}
else
{
  if (Math.Abs(FormZ) >= .9m && (Math.Abs(Form2Z) >= .9m ||
Math.Abs(Form2PrimeZ) >= .9m) && Math.Abs(EntropyZ) >= .9m)
  {
        SignificanceScore += 7000;
        Rule2Hit = true;
  }
}

// Rule Set 3
// ----------
// Only do this if Rule2 was not hit.
if (!Rule2Hit)
{
  if (Math.Abs(Form1_1Z) >= .9m || Math.Abs(Form1_2Z) >= .9m ||
Math.Abs(Form1_3Z) >= .9m || Math.Abs(Form1_4Z) >= .9m)
  {
        if (Math.Abs(Form1_1Z) >= .9m && (Math.Abs(Form2Z) >= .9m ||
Math.Abs(Form2PrimeZ) >= .9m))
        {
          SignificanceScore += 6000;
        }
        if (Math.Abs(Form1_2Z) >= .9m && (Math.Abs(Form2Z) >= .9m ||
Math.Abs(Form2PrimeZ) >= .9m))
        {
          SignificanceScore += 7000;
        }
```

```
            if (Math.Abs(Form1_3Z) >= .9m && (Math.Abs(Form2Z) >= .9m ||
Math.Abs(Form2PrimeZ) >= .9m))
            {
               SignificanceScore += 8000;
            }
            if (Math.Abs(Form1_4Z) >= .9m && (Math.Abs(Form2Z) >= .9m ||
Math.Abs(Form2PrimeZ) >= .9m))
            {
               SignificanceScore += 9000;
            }
   }
   else
   {
            if (Math.Abs(FormZ) >= .9m && (Math.Abs(Form2Z) >= .9m ||
Math.Abs(Form2PrimeZ) >= .9m))
            {
               SignificanceScore += 5000;
            }
   }

}

// Rule Set 4
// ----------
// Only do this if Rule2 was not hit.
if (!Rule2Hit)
{
   if (Math.Abs(Form1_1Z) >= .9m || Math.Abs(Form1_2Z) >= .9m ||
Math.Abs(Form1_3Z) >= .9m || Math.Abs(Form1_4Z) >= .9m)
   {
            if (Math.Abs(Form1_1Z) >= .9m && Math.Abs(EntropyZ) >= .9m)
            {
               SignificanceScore += 8000;
            }
            if (Math.Abs(Form1_2Z) >= .9m && Math.Abs(EntropyZ) >= .9m)
            {
               SignificanceScore += 9000;
            }
            if (Math.Abs(Form1_3Z) >= .9m && Math.Abs(EntropyZ) >= .9m)
            {
               SignificanceScore += 10000;
            }
```

```
            if (Math.Abs(Form1_4Z) >= .9m && Math.Abs(EntropyZ) >= .9m)
            {
               SignificanceScore += 11000;
            }
   }
   else
   {
            if (Math.Abs(FormZ) >= .9m && Math.Abs(EntropyZ) >= .9m)
            {
               SignificanceScore += 7000;
            }
   }
}

// Rule Set 5
// ----------
// Only do this if Rule2 was not hit.
if (!Rule2Hit)
{
   if ((Math.Abs(Form2Z) >= .9m || Math.Abs(Form2PrimeZ) >= .9m) &&
Math.Abs(EntropyZ) >= .9m)
   {
            SignificanceScore += 10000;
   }
}

// Rule Set 6
// ----------
if (Math.Abs(FractalZ) >= .9m)
{
   SignificanceScore += 10000;
}

// Rule Set 7
// ----------
SignificanceScore += BreakCoefficient * 5000;

// Rule Set 8
// ----------
if (RingThickness > 0 && Math.Abs(NormalizedAreaZ) >= .9m)
{
   SignificanceScore += 15000;
```

```
}
// Rule Set 9
// ----------
if (Form2Raw > Form1_1Raw)
{
   SignificanceScore += 20000;
}
// Rule Set 10
// ----------
if (Math.Abs(Form2Z) >= .9m && Math.Abs(Form2PrimeZ) >= .9m)
{
   SignificanceScore += 100000;
}
// Rule Set 11
// ----------
if (Ai4 > 0)
{
   SignificanceScore += 10000;
}
// Rule Set 12
// ----------
if (RingThickness == 0)
{
   SignificanceScore += 8000;
}
```

Once the significance score has been calculated, the final score can be determined using the following formulae:

EPIC Score Total Score = (EPIC Base Score + SignificanceScore) /100

The final step in determining the EPIC score is to perform a scaling with the range of 0 – 25, by applying the EPIC Total Score to the following transformation table:

EPIC Score Between	Score
0 - 99	**0**
100 - 199	**1**
200 - 299	**2**
300 - 399	**3**
400 - 499	**4**

EPIC Score Between	Score
500 - 599	5
600 - 699	6
700 - 799	7
800 - 899	8
900 - 999	9
1000 - 1099	10
1100 - 1199	11
1200 - 1349	12
1350 - 1449	13
1450 - 1549	14
1550 – 1649	15
1650 - 1799	16
1800 - 1899	17
1900 - 1999	18
2000 - 2149	19
2150 - 2299	20
2300 - 2449	21
2450 - 2599	22
2600 - 2749	23
2750 - 3000	24
3000+	25

Logistic Regression Score Calculation

The logistic regression score calculation relies on the same coefficient raw data described in the first part of this document, but performs radically different mathematical transformations in order to achieve a score. The score comprises of five (5) different factors that when added together and transformed create the base LR score.

The five factors are CV Factor, GI Factor, Liver Factor, Respiratory Factor and Kidney Factor. Each of these factors are calculated for each coefficient using the appropriate multiplier for the specific factor..

The factors that are used in the calculation can be found in the following document: EPIC Score Development Study.

The basic calculation to create the score involves performing a transformation (in some cases) on the raw data value and then multiplying it by a predetermined factor.

ClearView Scan Coefficient	Variable Name	Data Transformation
Age	Age	**None**
Average Intensity	AI	**None**
AI * Normalized Area	AI* NORMALIZEDAREA	**None**
Break	BREAKCOEFFICIENT	if BREAKCOEFFICIENT = 0 then BREAKBIN = 0 else if BREAKCOEFFICIENT > 0 then BREAKBIN = 1
Entropy	ENTROPY	**None**
Entropy * Form2	ENTROPY_LF2	LOG(FORM2)
Entropy * Form	ENTROPY_FORM1	LOG(FORM)
Fractal	FRACTALCOEFFICIENT	**None**
Intercept	INTERCEPT	**None**
Form	FORM1	LOG(FORM)
Form2	FORM2	LOG(FORM2)
Form * Form2	FORM_FORM2	LOG(FORM) * LOG(FORM2)
NS Integer	NSINTEGER	LOG(NSINTEGER + 5)
Normalized Area	NORMALIZEDAREA	**None**
Ring Intensity	RINGINTENSITY	**None**
Ring Thickness	RINGTHICKNESS	**None**
Male	MALE	**None**

- Only used on non-capacitive barrier images

The following calculation is done five times (once for each of the five organ systems, cardiovascular, gastrointestinal, renal, hepatic, and respiratory) for each image, replacing the multiplication factors with the appropriate values for each of the five factor groups. The code below implements the calculations described in the table above. Specific variables in the equation are normalized before use.

```
AgeFactor = (SubjectAge * IrfactorsDmEntity.Age);
AiFactor = (Convert.ToDecimal(ratingVar.AverageIntensity.Value) *
IrfactorsDmEntity.Ai);
Ai_NaFactor = ((Convert.ToDecimal(ratingVar.AverageIntensity.Value) *
Convert.ToDecimal(ratingVar.NormalizedArea.Value))* IrfactorsDmEntity.AiNa);
BreakBinFactor = (Convert.ToDecimal(BreakVar * IrfactorsDmEntity.BreakBin));
```

```
EntropyFactor = (Convert.ToDecimal(ratingVar.Entropy.Value) * lrfactorsDmEntity.Ent);
Entropy_Lf2Factor = ((Convert.ToDecimal(ratingVar.Entropy.Value) *
Convert.ToDecimal(Math.Log(ratingVar.Form2.Value))) * lrfactorsDmEntity.EntLf2);
Entropy_LformFactor = ((Convert.ToDecimal(ratingVar.Entropy.Value) *
Convert.ToDecimal(Math.Log(ratingVar.FormCoefficient.Value))) *
lrfactorsDmEntity.EntLform);
FractalFactor = (Convert.ToDecimal(ratingVar.FractalCoefficient.Value) *
lrfactorsDmEntity.Fr);
InterceptFactor = (lrfactorsDmEntity.Intercept);
Lf2Factor = (Convert.ToDecimal(Math.Log(ratingVar.Form2.Value)) *
lrfactorsDmEntity.Lf2);
LformFactor = (Convert.ToDecimal(Math.Log(ratingVar.FormCoefficient.Value)) *
lrfactorsDmEntity.Lform);
if (!IsFiltered)
   Lform_Lf2Factor = ((Convert.ToDecimal(Math.Log(ratingVar.FormCoefficient.Value))
* Convert.ToDecimal(Math.Log(ratingVar.Form2.Value))) * lrfactorsDmEntity.LformLf2);
else
   Lform_Lf2Factor = 0;
LnsFactor = (Convert.ToDecimal(Math.Log(ratingVar.JsInteger.Value + 5.0)) *
lrfactorsDmEntity.Lns);
MaleFactor = (MaleVar * lrfactorsDmEntity.Male);
NaFactor = (Convert.ToDecimal(ratingVar.NormalizedArea.Value) *
lrfactorsDmEntity.Na);
RinFactor = (Convert.ToDecimal(ratingVar.RingIntensity.Value) *
lrfactorsDmEntity.Rin);
RthFactor = (Convert.ToDecimal(ratingVar.RingThickness.Value) *
lrfactorsDmEntity.Rth);

TotalScoreValue = AgeFactor + AiFactor + Ai_NaFactor + BreakBinFactor +
EntropyFactor + Entropy_Lf2Factor + Entropy_LformFactor + FractalFactor +
InterceptFactor + Lf2Factor + LformFactor + Lform_Lf2Factor + LnsFactor +
          MaleFactor + NaFactor + RinFactor + RthFactor;

TransformedTotalScore =
Convert.ToDecimal(Math.Exp(Convert.ToDouble(TotalScoreValue)) / (1 +
Math.Exp(Convert.ToDouble(TotalScoreValue))));
```

The total score is value is obtained by adding together the 5 totals that were created, one for each of the factors for the given sector.

Once that total is obtained, the following formula is used to convert the value to the final LR score for the given sector:

LR Score =((Summation of the 5 scores) / 3.5) * 35

* If the value is above 25 it is set at 25.

This process is repeated for all sectors and for all 20 images.

Naive/Bayes Score Calculation

The NB score calculation relies on the same raw data described in the first part of this document, but performs radically different mathematical transformations in order to achieve a score. The score comprises of five (5) different factors that when added together and transformed create the NB score.

The calculation relies on a total of 1482 a set of pre-calculated static factors that are used in the calculation.

The values used in the calculation were developed and the development was documented in an engineering study, EPIC ClearView Algorithm Development Report, ENG-025.

The transformations used in the calculations are below:

ClearView Scan Coefficient	Variable Name	Data Transformation
Average Intensity	AI	**None**
Average Intensity (1)	AI1	**None**
Average Intensity (2)	AI2	**None**
Average Intensity (3)	AI3	LOG(AI3 + 5)
Average Intensity (4)	AI4	LOG(AI4 + 5)
Break	BREAKCOEFFICIENT	**None**
Entropy	ENTROPY	**None**
Fractal	FRACTALCOEFFICIENT	**None**
Form	FORM1	LOG(FORM)
Form (1)	FORM1_1	LOG(FORM1_1)
Form (2)	FORM1_2	LOG(FORM1_2)
Form (3)	FORM1_3	LOG(FORM1_3 + 5)
Form (4)	FORM1_4	LOG(FORM1_4 + 5)
Form 2	FORM2	LOG(FORM2)
Form 2 Prime	FORM2PRIME	LOG(FORM2PRIME)
NS Integer	NSINTEGER	LOG(NSINTEGER + 5)

ClearView Scan Coefficient	Variable Name	Data Transformation
Normalized Area	NORMALIZEDAREA	**None**
Ring Intensity	RINGINTENSITY	**None**
Ring Thickness	RINGTHICKNESS	**None**

The formula used to calculate the score value for each of the five factors (CV, GI ….) is shown below, this is done for each of the 5 factors to obtain 5 totals:

```
ScoreValue = ((ar.Ai1 * CoefficientsToUse.AI1_Factor) +
             (ar.AverageIntensity * CoefficientsToUse.AI_Factor) +
             (Math.Log(ar.Form2Prime) * CoefficientsToUse.LF2P_Factor) +
             (Math.Log(ar.FormCoefficient) * CoefficientsToUse.LFORM_Factor) +
             (ar.Ai2 * CoefficientsToUse.AI2_Factor) +
             (Math.Log(ar.Ai3 + 5.0) * CoefficientsToUse.AI3_Factor) +
             (Math.Log(ar.Ai4 + 5.0) * CoefficientsToUse.AI4_Factor) +
             (BreakVar * CoefficientsToUse.BREAKBIN_Factor) +
             (ar.Entropy * CoefficientsToUse.ENT_Factor) +
             (ar.FractalCoefficient * CoefficientsToUse.FR_Factor) +
             (Math.Log(ar.Form11) * CoefficientsToUse.LF11_Factor) +
             (Math.Log(ar.Form12) * CoefficientsToUse.LF12_Factor) +
             (Math.Log(ar.Form13 + 5.0) * CoefficientsToUse.LF13_Factor) +
             (Math.Log(ar.Form14 + 5.0) * CoefficientsToUse.LF14_Factor) +
             (Math.Log(ar.Form2) * CoefficientsToUse.LF2_Factor) +
             (Math.Log(ar.JsInteger + 5.0) * CoefficientsToUse.LNS_Factor) +
             (ar.RingIntensity * CoefficientsToUse.RIN_Factor) +
             (ar.RingThickness * CoefficientsToUse.RTH_Factor) +
             (ar.NormalizedArea * CoefficientsToUse.SA_Factor));
```

Once the five totals are obtained and totaled for all sectors (loop through 78 times), they are multiplied by the respective values below and added together, additionally the intercept value is also added into the score:

Cap Barrier	System	Intercept	CV	GI	LIV	RESP	KID
No	CV	-6.91	-3.43	0	-2.03	0	11.54
No	RESP	-3.72	2.58	-13.53	-4.05	5.79	13.97
No	LIV	-4.42	0	-7.85	-3.3	0	12.63
No	GI	-10.426	-29.41	33.35	8.97	16.08	-22.88
No	KID	-12.54	-38.64	30.71	12.39	34.98	-26.59
Yes	CV	5.88	-5.07	2.03	-5.06	3.17	10.3

Cap Barrier	System	Intercept	CV	GI	LIV	RESP	KID
Yes	RESP	1.82	-7.25	2.19	-6.9	7.59	7.55
Yes	LIV	-.82	-14.73	13.95	11.87	0	-3.55
Yes	GI	-4.83	-11.24	13.38	-1.07	1.79	-1.37
Yes	KID	2.18	-9.6	2.54	-4.64	7.95	8.93

The following code generates the reported NB score for the given sector. This function takes the values generated for each of the organ systems and performs the final transformation to scale the result to a value between 0 and 25.

```
TempScore = (CVValue * Convert.ToDouble(Coefficients.Nbcvcoeff)) +
        (GIValue * Convert.ToDouble(Coefficients.Nbgicoeff)) +
        (LIVValue * Convert.ToDouble(Coefficients.Nblivcoeff)) +
        (RESPValue * Convert.ToDouble(Coefficients.Nbrespcoeff)) +
        (KIDValue * Convert.ToDouble(Coefficients.Nbkidcoeff)) +
        Convert.ToDouble(Coefficients.InterceptCoeff);
TempScore = Math.Exp(TempScore) / (1 + Math.Exp(TempScore)) *
Convert.ToDouble(Coefficients.ScoreMultiplierFactor);
// Sanity checks
if (TempScore < 0)
   TempScore = 0;
else if (TempScore > 25)
   TempScore = 25;
return Convert.ToInt32(Math.Round((TempScore), 0));
```

This score is then stored in the database.

Report Score Selection

The NB scores are directly presented on the report. They are displayed with a set of odds ratios (development of odds ratios values are documented in an engineering study, EPIC ClearView Algorithm Development Report, ENG-025. that help to explain the strength of association between the ClearView Response Scale and the presence of a disease state. The odds ratios are fixed values that were determined based on initial population analysis and do not change unless the development model changes.

The table below represents the current odds ratios:

Component	Capacitive Barrier	Non-capacitive Barrier
CV	3.9	2.8
RESP	7.8	2.7
LIV	13.9	6.8
GI	45.5	8.6
KID	99.1	3.2

The process of determining which score to display between the EPIC score and the LR score is outlined below:

If the EPIC Score is larger, display it.

If the LR Score is larger, perform the following transformation(s) and display the resultant score:

New Score = (EPIC Score + LR Score)/2.0

If the difference between the EPIC Score and the LR Score is >= 10 then add 4 to New Score

Use New Score

The reported score (either EPIC score or New Score) is saved into the database to be pulled by the report process for display.

Report Production

The reported scores are pulled from the table in the database where they were stored and displayed in the report tabs of the ClearView Software Solution. The specifications for the report tabs are captured in the Analysis Functions Specifications (SS-205).

4.0 Reference Documents

SS-203-01, Finger Sector Map
ENG-026, EPIC Score Development Study
SS-205, Analysis Functions Specifications
ENG-025, EPIC ClearView Algorithm Development Report

Document Revision History

Version Number:	Description of Change:	Date:	Updated by:
000	Introduction	3/5/12	A. Mason