GrahamGroup

October 30, 2014

1 Robust Extraction of Quantitative Information from Histology Images

Quentin Caudron Romain Garnier ***with Bryan Grenfell and Andrea Graham***

1.0.1 Outline

- Image processing
- Extracted measures
- Preliminary analysis
- Future directions

4. Age as random effect <—

["interface_hepatitis", "confluent_necrosis", "portal_inflammation", "ln_ap_ri"]

```
In [3]: def normalise(df, skip = []) :
                for i in df.columns :
                        if i not in skip :
                                df[i] -= df[i].mean()
                                df[i] /= df[i].std()
                return df
```

```
        def rescale(df, skip = []) :
            for i in df.columns :
                if i not in skip :
                    df[i] -= df[i].min()
                    df[i] /= df[i].max()
            return df
```

```
        # Remove a layer from a list
        def delayer(m) :
                out = []
                for i in m :
                        if isinstance(i, list) :
                                for j in i :
```

```
                                out.append(j)
                        else :
                                out.append(i)
        return out
```

```
# Remove all layers from a list
def flatten(m) :
        out = m[:]

        while out != delayer(out) :
                out = delayer(out)

        return out
```

```
# Generate all combinations of objects in a list
def combinatorial(l) :
        out = []

        for numel in range(len(l)) :
                for i in itertools.combinations(l, numel+1) :
                        out.append(list(i))

        return out
```

```
def pcaplot(df) :

        # PCA
        pca = decomposition.PCA(whiten = True)
        pca.fit(df)
        p1 = pca.components_[0] / np.abs(pca.components_[0]).max() * np.sqrt(2)/2
        p2 = pca.components_[1] / np.abs(pca.components_[1]).max() * np.sqrt(2)/2
```

```
        # Normalise
        norms = np.max([np.sqrt((np.array(zip(p1, p2)[i])**2).sum()) for i in range(len(p1))])
        c = plt.Circle( (0, 0), radius = 1, alpha = 0.2)
        plt.axes(aspect = 1)
        plt.gca().add_artist(c)

        plt.scatter(p1 / norms, p2 / norms)
        plt.xlim([-1, 1])
        plt.ylim([-1, 1])

        for i, text in enumerate(df.columns) :
                plt.annotate(text, xy = [p1[i], p2[i]])

        plt.tight_layout()
```

```
def test_all_linear(df, y, x, return_significant = False, group = None) :

    # All possible combinations of independent variables
        independent = combinatorial(x)

        fits = {}
        pval = {}
        linmodels = {}
        qsum = {}
        aic = {}

        # For all dependent variables, one at a time
        for dependent in y :

                print "Fitting for %s." % dependent

                # For all combinations of independent variables
                for covariate in independent :

                        # Standard mixed model
                        if group is None :

                                # Fit a linear model
                                subset = delayer([covariate, dependent])
                                df2 = df[delayer(subset)].dropna()
                                df2["Intercept"] = np.ones(len(df2))

                                ols = sm.GLS(endog = df2[dependent], exog = df2[delayer([covaria
```

```
                                # Save the results
                                if (return_significant and ols.f_pvalue < 0.05) or (not return_s
                                        linmodels.setdefault(dependent, []).append(ols)
                                        fits.setdefault(dependent, []).append(ols.rsquared)
                                        pval.setdefault(dependent, []).append(ols.f_pvalue)
                                        aic.setdefault(dependent, []).append(ols.aic)


                        # Mixed effects model
                        else :
                                subset = delayer([covariate, dependent, group])
                                df2 = df[delayer(subset)].dropna()

                                # Fit a mixed effects model
                                ols = MixedLM(endog = df2[dependent], exog = df2[covariate], gr

                                # Calculate AIC
                                linmodels.setdefault(dependent, []).append(ols)
                                fits.setdefault(dependent, []).append(2 * (ols.k_fe + 1) - 2 *
                                pval.setdefault(dependent, []).append(ols.pvalues)

        if group is not None :
                for i in y :
                        f = np.array(fits[i])
                        models = np.array(linmodels[i])
                        idx = np.where(f - f.min() <= 2)[0]
                        bestmodelDoF = [j.k_fe for j in np.array(linmodels[i])[idx]]
                        bestmodels = [idx[j] for j in np.where(bestmodelDoF == np.min(bestmodell
                        qsum[i] = models[idx[np.where(f[bestmodels] == np.min(f[bestmodels]))]]


                return linmodels, fits, pval, qsum

        return linmodels, fits, pval, aic









def summary(models) :
```

```
        # Generate list of everything
        r2 = np.array([m.r2 for dependent in models.keys() for m in models[dependent]])
        p = np.array([m.f_stat["p-value"] for dependent in models.keys() for m in models[depend
        mod = np.array([m for dependent in models.keys() for m in models[dependent]])
        dependent = np.array([dependent for dependent in models.keys() for m in models[dependen

        # Sort by R2
        idx = np.argsort(r2)[::-1]

        # Output string
        s = "%d significant regressions.\n\n" % len(r2)
        s += "Ten most correlated :\n\n"

        # Print a summary of the top ten correlations
        for i in idx[:10] :
                s += ("%s ~ %s\n" % (dependent[i], " + ".join(mod[i].x.columns[:-1])))
                s += ("R^2 = %f\tp = %f\n\n" % (r2[i], p[i]))

        print s
```

```
    def rstr(y, x) :
        formatstr = "%s ~ " % y
        for i in x[:-1] :
            formatstr += str(i)
            formatstr += " + "
        formatstr += str(x[-1])
        return formatstr
```

In [14]:
```
import numpy as np
from sklearn.neighbors import KernelDensity
from matplotlib import rcParams
import matplotlib.pyplot as plt
import seaborn
import pandas as pd
import itertools
from sklearn import linear_model, ensemble, decomposition, cross_validation, preprocessing
from statsmodels.regression.mixed_linear_model import MixedLM
import statsmodels.api as sm
from statsmodels.regression.linear_model import OLSResults
from statsmodels.tools.tools import add_constant


%matplotlib inline
rcParams["figure.figsize"] = (14, 8)


# RAW DATA
```

```
raw_physical = pd.read_csv("../data/physical.csv")
raw_histo = pd.read_csv("../data/tawfik.csv")
ent = pd.read_csv("../4x/results/entropy.csv").drop(["Unnamed: 0"], 1)
foci = pd.read_csv("../4x/results/foci.csv").drop(["Unnamed: 0"], 1)
lac = pd.read_csv("../4x/results/normalised_lacunarity.csv").drop(["Unnamed: 0"], 1)
gabor = pd.read_csv("../4x/results/gabor_filters.csv").drop(["Unnamed: 0"], 1)
ts = pd.read_csv("../4x/results/tissue_sinusoid_ratio.csv").drop(["Unnamed: 0"], 1)

raw_image = pd.merge(lac, ent,
        on=["Sheep", "Image"]).merge(foci,
        on=["Sheep", "Image"]).merge(gabor,
        on=["Sheep", "Image"]).merge(ts,
    on=["Sheep", "Image"])
raw_image.rename(columns = {        "meanSize" : "FociSize",
                                                                "TSRatio" : "TissueToSinusoid"
                                                                "Count" : "FociCount" }, inpla



# CLEAN DATA

physcols = ["Weight", "Sex", "AgeAtDeath", "Foreleg", "Hindleg"]
imagecols = ["Entropy", "Lacunarity", "Inflammation", "Scale", "Directionality", "FociCount",
histcols = ["Lobular_collapse", "Interface_hepatitis", "Confluent_necrosis", "Ln_ap_ri", "Porta




# IMAGE

# Set FociSize to zero if FociCount is zero
# Drop stdSize
image = raw_image
image = image.drop("stdSize", 1)
image.FociSize[raw_image.FociCount == 0] = 0



# HISTO

histo = raw_histo
histo = histo.drop(["Vessels", "Vacuol", "Pigment", "Std_hep_size"], 1)



# PHYSICAL

physical = raw_physical
physical = physical.drop(["CurrTag", "DeathDate", "Category"], 1)
physical
```

```
# COMPLETE DATASET

raw_data = pd.merge(pd.merge(image, histo, on="Sheep", how="outer"), physical, on="Sheep", how=
raw_data.to_csv("../data/tentative_complete.csv")



# AVERAGED BY SHEEP
data = raw_data
data["Inflammation"] = data.FociCount * data.FociSize

sheep = rescale(data.groupby("Sheep").mean())
age = rescale(data.groupby("AgeAtDeath").mean())




# REGRESSIONS : fixed effects, grouped by sheep

df = sheep[["Portal_inflammation", "FociSize"]].dropna()
df["Intercept"] = np.ones(len(df))
portal_inflammation = sm.GLS(endog = df.Portal_inflammation, exog = df[["FociSize", "Intercept
#portal_inflammation = portal_inflammation.summary()
del portal_inflammation.tables[2]



df = sheep[["BD_hyperplasia", "Scale", "Directionality", "FociSize"]].dropna()
df["Intercept"] = np.ones(len(df))
hyperplasia = sm.GLS(endog = df.BD_hyperplasia, exog = df[["FociSize", "Scale", "Directionalit
#hyperplasia.summary()
del hyperplasia.tables[2]




# REGRESSIONS : fixed effects, grouped by age

df = age[["Max_hep_size", "Entropy", "Directionality"]].dropna()
df["Intercept"] = np.ones(len(df))
maxhepsize = sm.GLS(endog = df.Max_hep_size, exog = df[["Entropy", "Directionality", "Intercep
del maxhepsize.tables[2]



df = age[["Lobular_collapse", "FociSize"]].dropna()
```

```
df["Intercept"] = np.ones(len(df))
lobular_collapse = sm.GLS(endog = df.Lobular_collapse, exog = df[["FociSize", "Intercept"]]).fi
del lobular_collapse.tables[2]


df = age[["Interface_hepatitis", "Lacunarity"]].dropna()
df["Intercept"] = np.ones(len(df))
interface_hepatitis = sm.GLS(endog = df.Interface_hepatitis, exog = df[["Lacunarity", "Intercep
del interface_hepatitis.tables[2]


df = age[["Fibrosis", "Inflammation"]].dropna()
df["Intercept"] = np.ones(len(df))
fibrosis = sm.GLS(endog = df.Fibrosis, exog = df[["Inflammation", "Intercept"]]).fit().summary
del fibrosis.tables[2]




# PCA

s = sheep.dropna(subset=delayer([imagecols, histcols]))
pca = decomposition.PCA(n_components=1)
pcax = pca.fit_transform(s[imagecols])
pcay = pca.fit_transform(s[histcols])
pca = sm.GLS(endog = pcay[:, 0][:, np.newaxis], exog = add_constant(pcax)).fit().summary()
del pca.tables[2]




# REGRESSIONS : mixed effects, intercept on age at death

df = age[["Fibrosis", "Inflammation"]].dropna()
df["Intercept"] = np.ones(len(df))
fibrosis = sm.GLS(endog = df.Fibrosis, exog = df[["Inflammation", "Intercept"]]).fit().summary
del fibrosis.tables[2]
```

In [19]:
```
a = portal_inflammation.summary()
del a.tables[2]
a
```

Out[19]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            GLS Regression Results
==============================================================================
Dep. Variable:     Portal_inflammation   R-squared:                       0.280
Model:                             GLS   Adj. R-squared:                  0.273
Method:                  Least Squares   F-statistic:                     37.34
Date:                 Tue, 28 Oct 2014   Prob (F-statistic):           2.12e-08
Time:                         23:35:30   Log-Likelihood:                 14.996
No. Observations:                   98   AIC:                            -25.99
Df Residuals:                       96   BIC:                            -20.82
Df Model:                            1
```

```
Covariance Type:             nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
FociSize       0.5627      0.092      6.111      0.000          0.380     0.746
Intercept      0.3368      0.043      7.855      0.000          0.252     0.422
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
"""
```

1.1 Image Processing

1.1.1 Extraction

- Automagical
- Reasonably quick

1.1.2 Robust

- Invariant to staining, slicing, field-related variation
- Capture intersample variation

1.2 Structural and Textural Measures

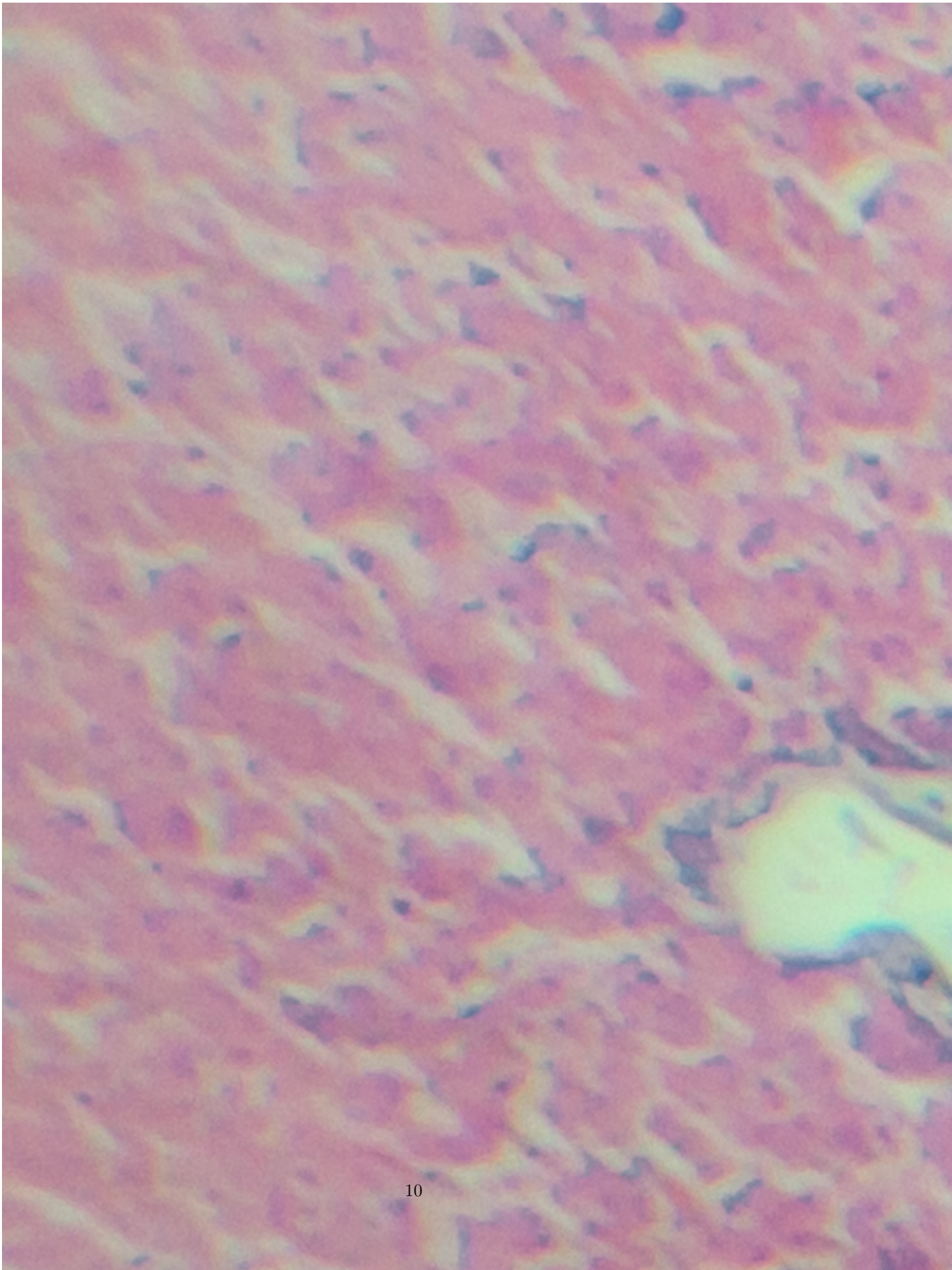
- characteristic **scale** of sinusoid widths
- **directional** amplitude of preferred sinusoid alignment
- **tissue to sinusoid** ratio
- **count** of inflammatory foci per image
- **mean size** of inflammatory foci per image
- information **entropy** of sinusoid distribution
- **lacunarity** (clustering) of sinusoids

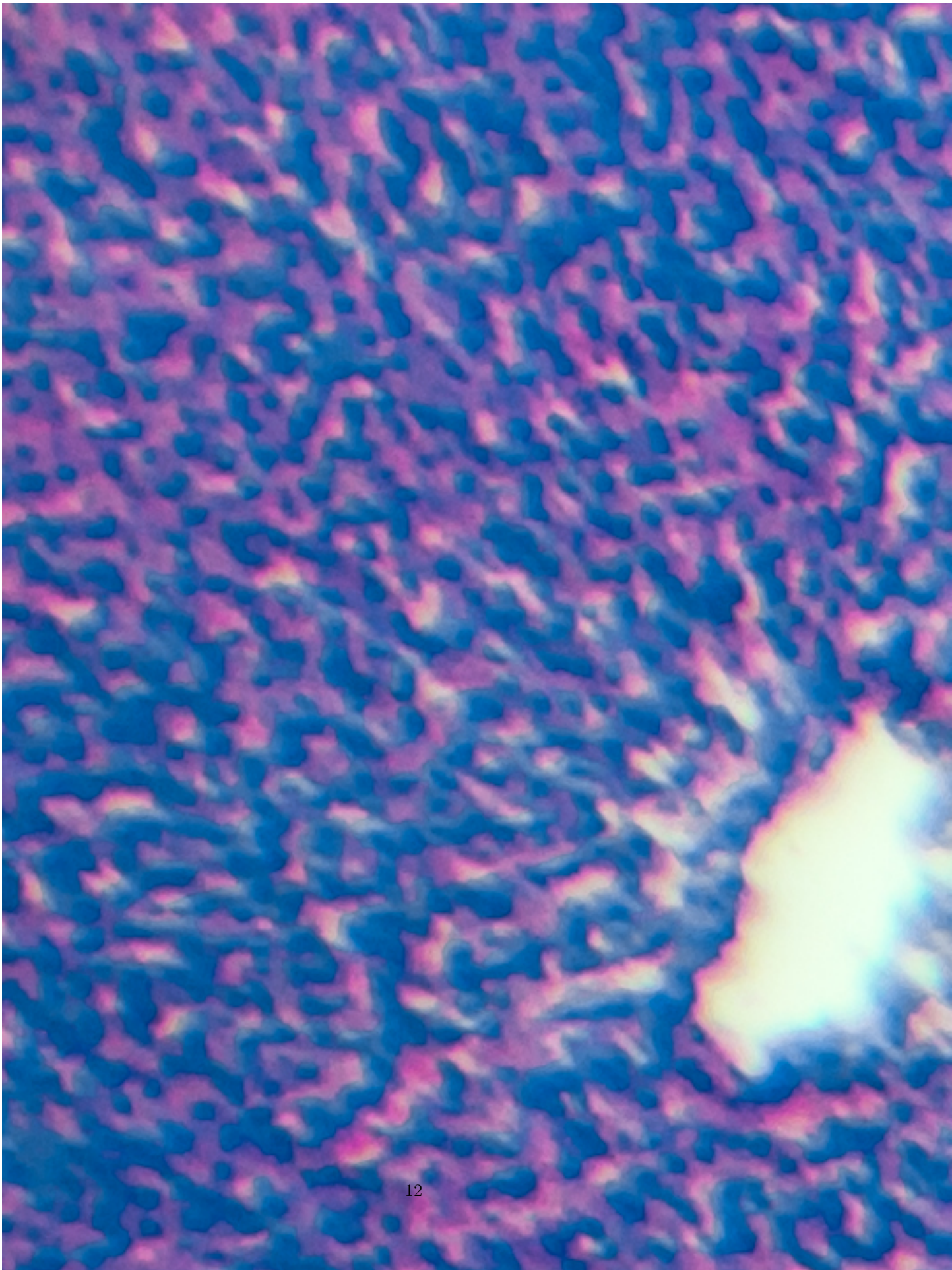
1.3 Exploratory Analysis

1.3.1 by individual

In [29]:
```
portal_inflammation
```

Out[29]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            GLS Regression Results
==============================================================================
Dep. Variable:     Portal_inflammation   R-squared:                       0.280
Model:                             GLS   Adj. R-squared:                  0.273
Method:                  Least Squares   F-statistic:                     37.34
Date:                 Tue, 28 Oct 2014   Prob (F-statistic):           2.12e-08
Time:                         23:40:10   Log-Likelihood:                 14.996
No. Observations:                   98   AIC:                            -25.99
Df Residuals:                       96   BIC:                            -20.82
Df Model:                            1
Covariance Type:             nonrobust
==============================================================================
```





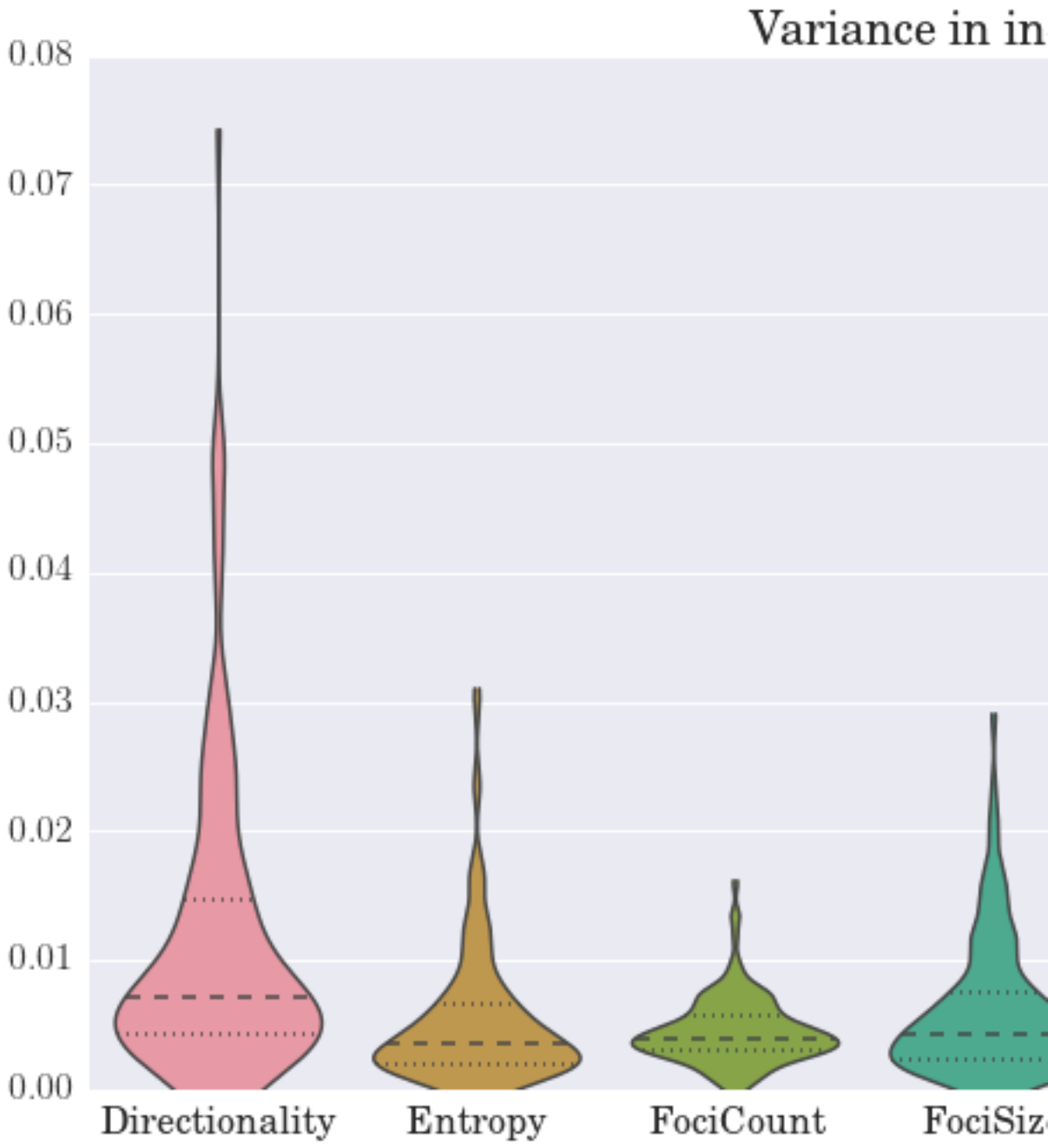


Figure 5: image

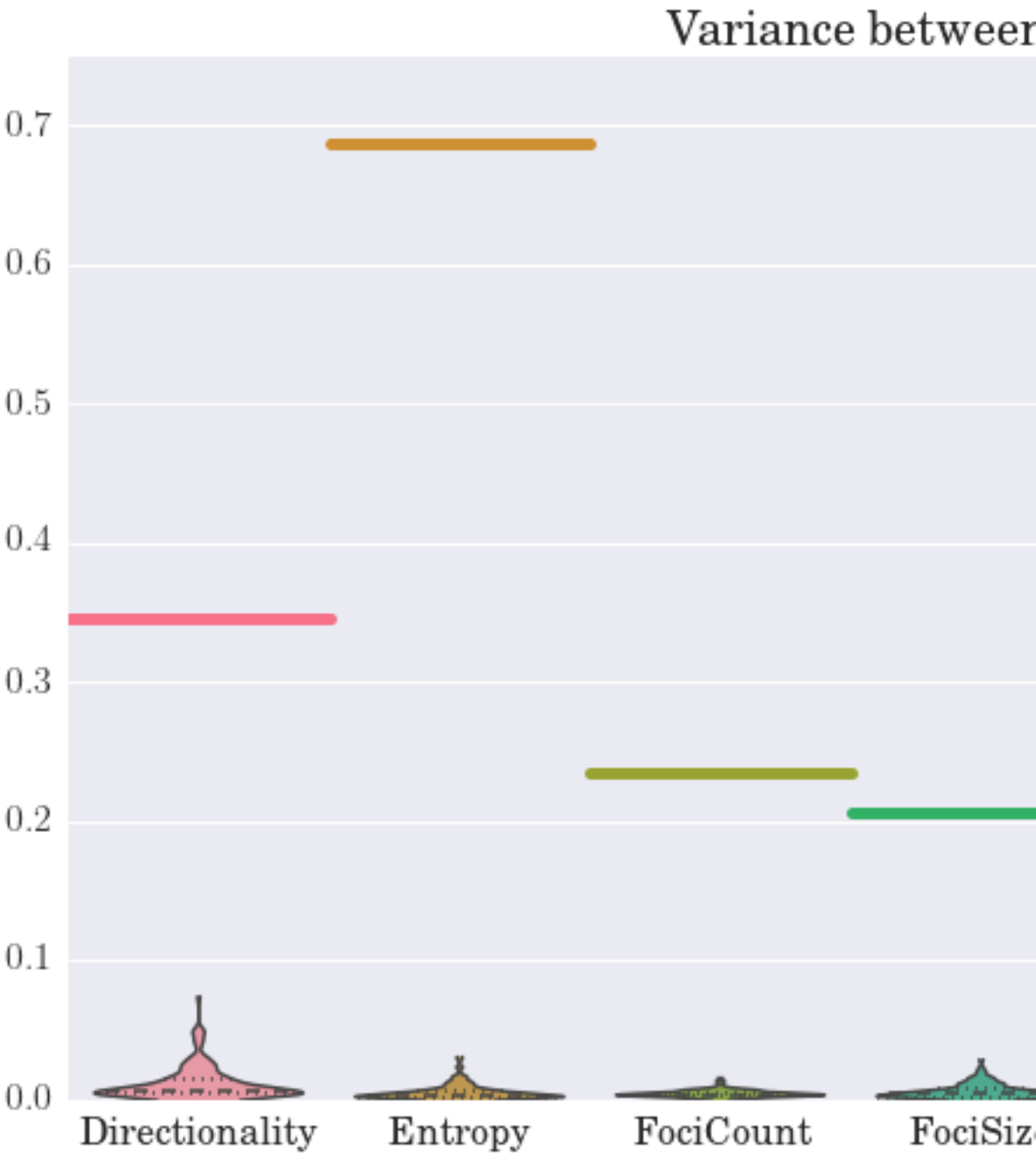


Figure 6: image

```
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
-------------------------------------------------------------------------------
FociSize       0.5627      0.092      6.111      0.000         0.380     0.746
Intercept      0.3368      0.043      7.855      0.000         0.252     0.422
===============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
"""
```

In [31]: hyperplasia

Out[31]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            GLS Regression Results
==============================================================================
Dep. Variable:         BD_hyperplasia   R-squared:                       0.306
Model:                            GLS   Adj. R-squared:                  0.284
Method:                 Least Squares   F-statistic:                     13.83
Date:                Tue, 28 Oct 2014   Prob (F-statistic):           1.52e-07
Time:                        23:40:10   Log-Likelihood:                -3.9632
No. Observations:                  98   AIC:                             15.93
Df Residuals:                      94   BIC:                             26.27
Df Model:                           3
Covariance Type:            nonrobust
===============================================================================
                     coef    std err          t      P>|t|      [95.0% Conf. Int.]
-------------------------------------------------------------------------------
FociSize           0.6698      0.113      5.902      0.000         0.444     0.895
Scale              0.5811      0.243      2.394      0.019         0.099     1.063
Directionality    -0.4419      0.190     -2.330      0.022        -0.819    -0.065
Intercept         -0.0504      0.079     -0.642      0.523        -0.206     0.105
===============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
"""
```

In [15]: pca

Out[15]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            GLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.075
Model:                            GLS   Adj. R-squared:                  0.065
Method:                 Least Squares   F-statistic:                     7.723
Date:                Wed, 29 Oct 2014   Prob (F-statistic):            0.00657
Time:                        14:38:47   Log-Likelihood:                -70.082
No. Observations:                  97   AIC:                             144.2
Df Residuals:                      95   BIC:                             149.3
Df Model:                           1
Covariance Type:            nonrobust
```

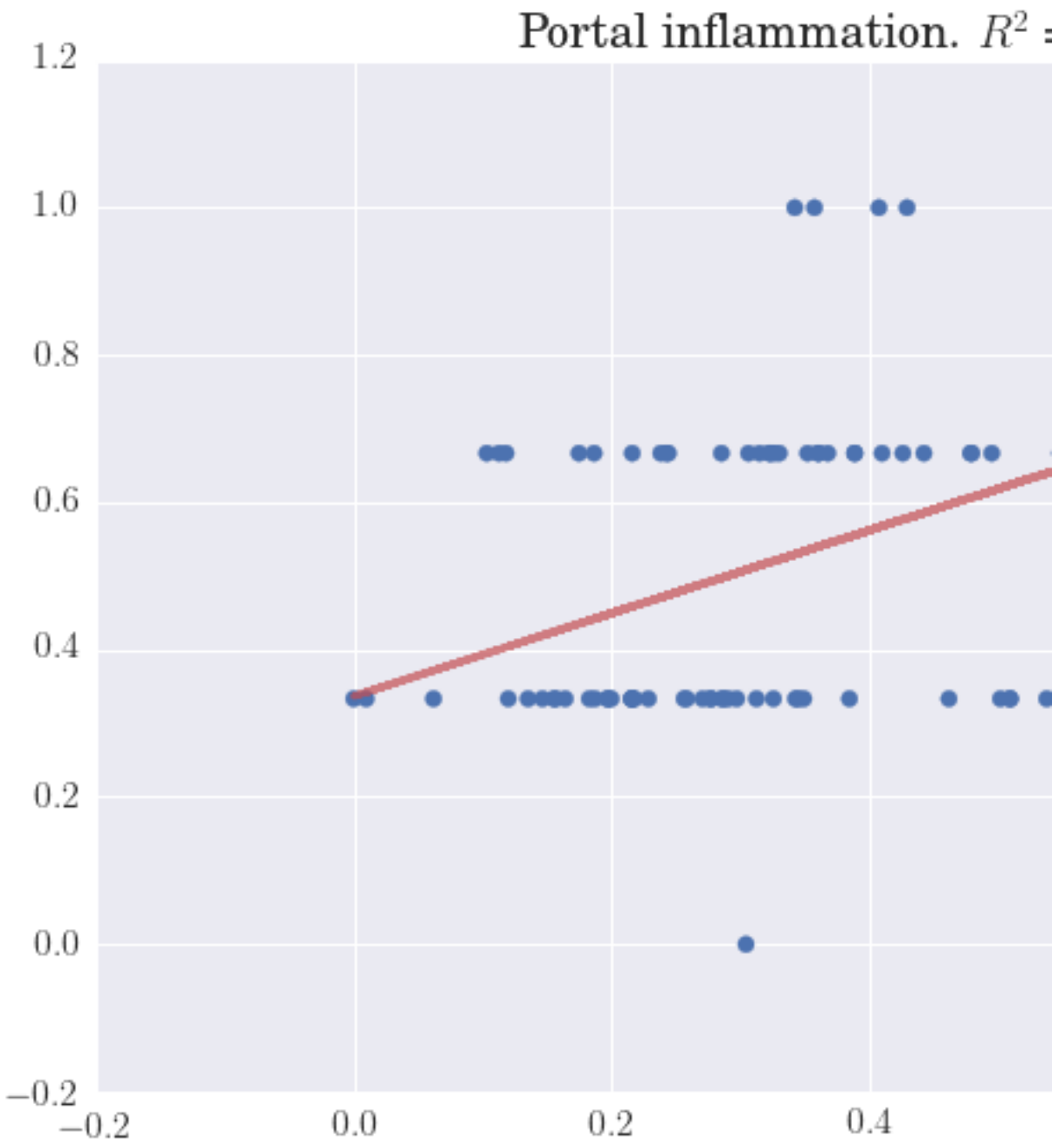


Figure 7: image

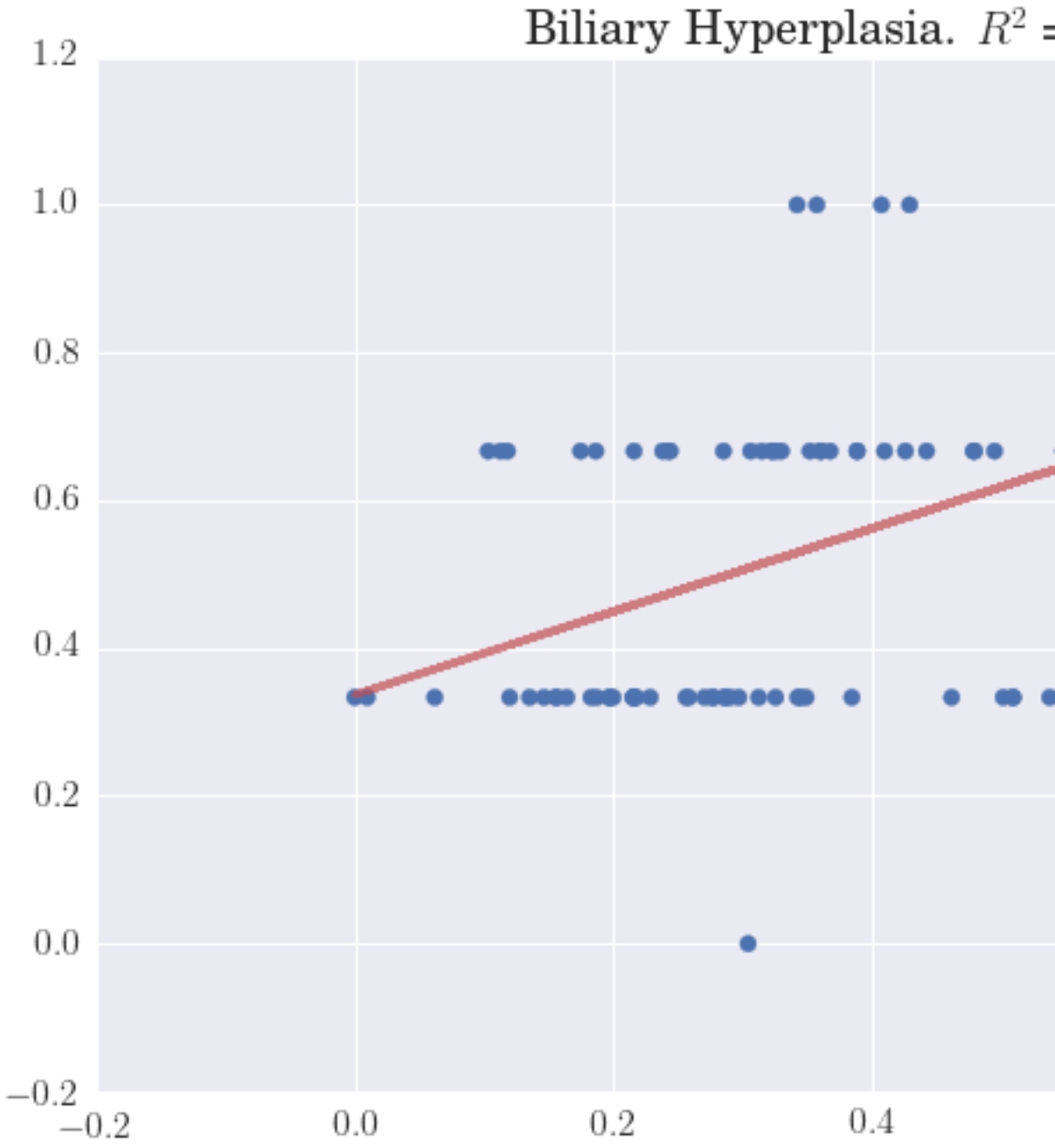


Figure 8: image

```
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
const      -2.949e-17      0.051  -5.77e-16      1.000        -0.102     0.102
x1             0.3865      0.139      2.779      0.007         0.110     0.663
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
"""
```

1.4 Exploratory Analysis

1.4.1 by age class

In [6]:
```
fibrosis
```

Out[6]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            GLS Regression Results
==============================================================================
Dep. Variable:               Fibrosis   R-squared:                       0.800
Model:                            GLS   Adj. R-squared:                  0.778
Method:                 Least Squares   F-statistic:                     36.07
Date:                Wed, 29 Oct 2014   Prob (F-statistic):           0.000201
Time:                        11:13:48   Log-Likelihood:                 7.8003
No. Observations:                  11   AIC:                            -11.60
Df Residuals:                       9   BIC:                            -10.80
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
Inflammation     1.0159      0.169      6.006      0.000         0.633     1.399
Intercept       -0.0105      0.083     -0.126      0.902        -0.198     0.177
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
"""
```

In [7]:
```
lobular_collapse
```

Out[7]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            GLS Regression Results
==============================================================================
Dep. Variable:       Lobular_collapse   R-squared:                       0.586
Model:                            GLS   Adj. R-squared:                  0.540
Method:                 Least Squares   F-statistic:                     12.73
Date:                Wed, 29 Oct 2014   Prob (F-statistic):            0.00605
Time:                        11:13:48   Log-Likelihood:                 2.2626
No. Observations:                  11   AIC:                           -0.5252
Df Residuals:                       9   BIC:                            0.2706
```

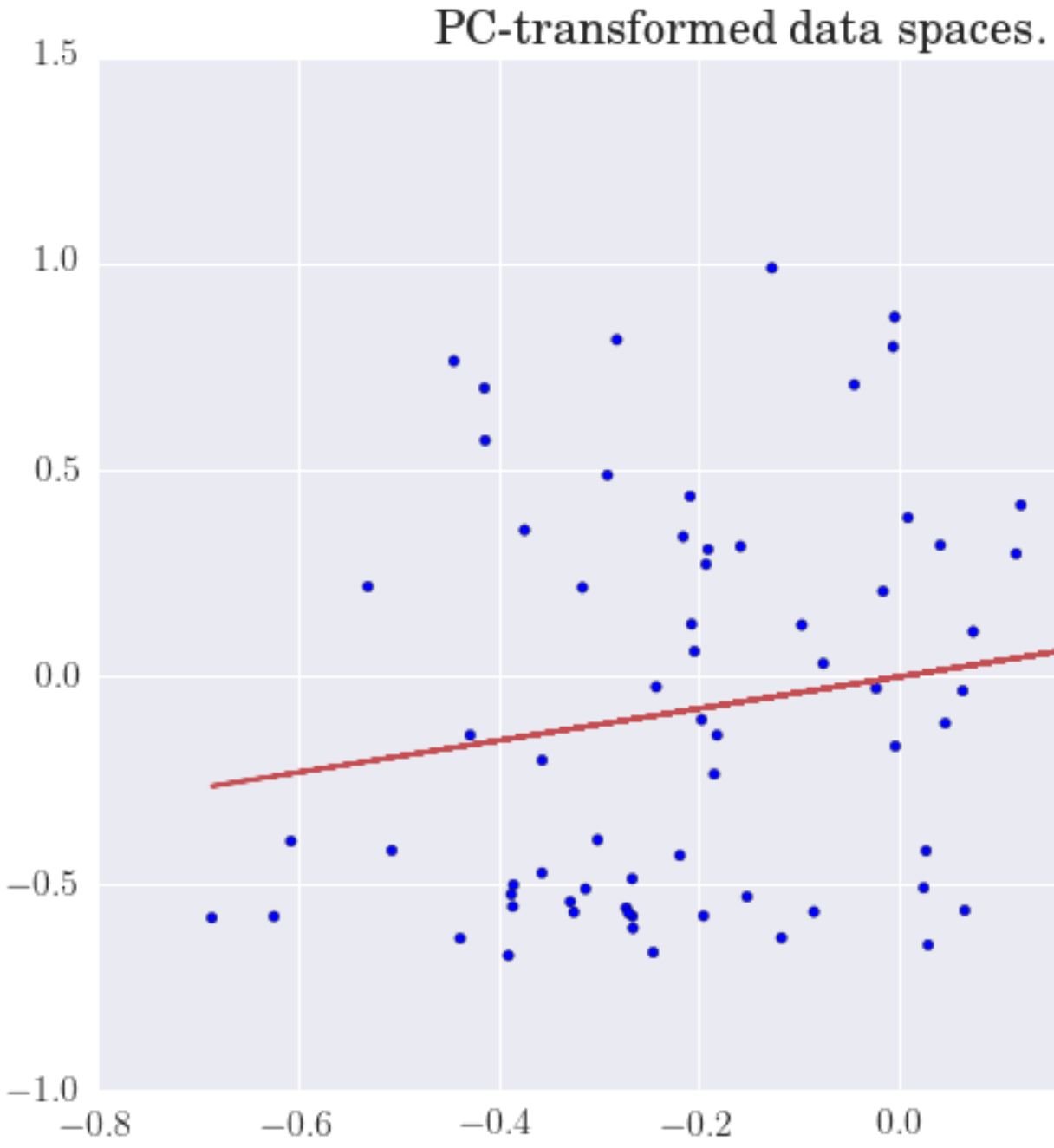


Figure 9: image

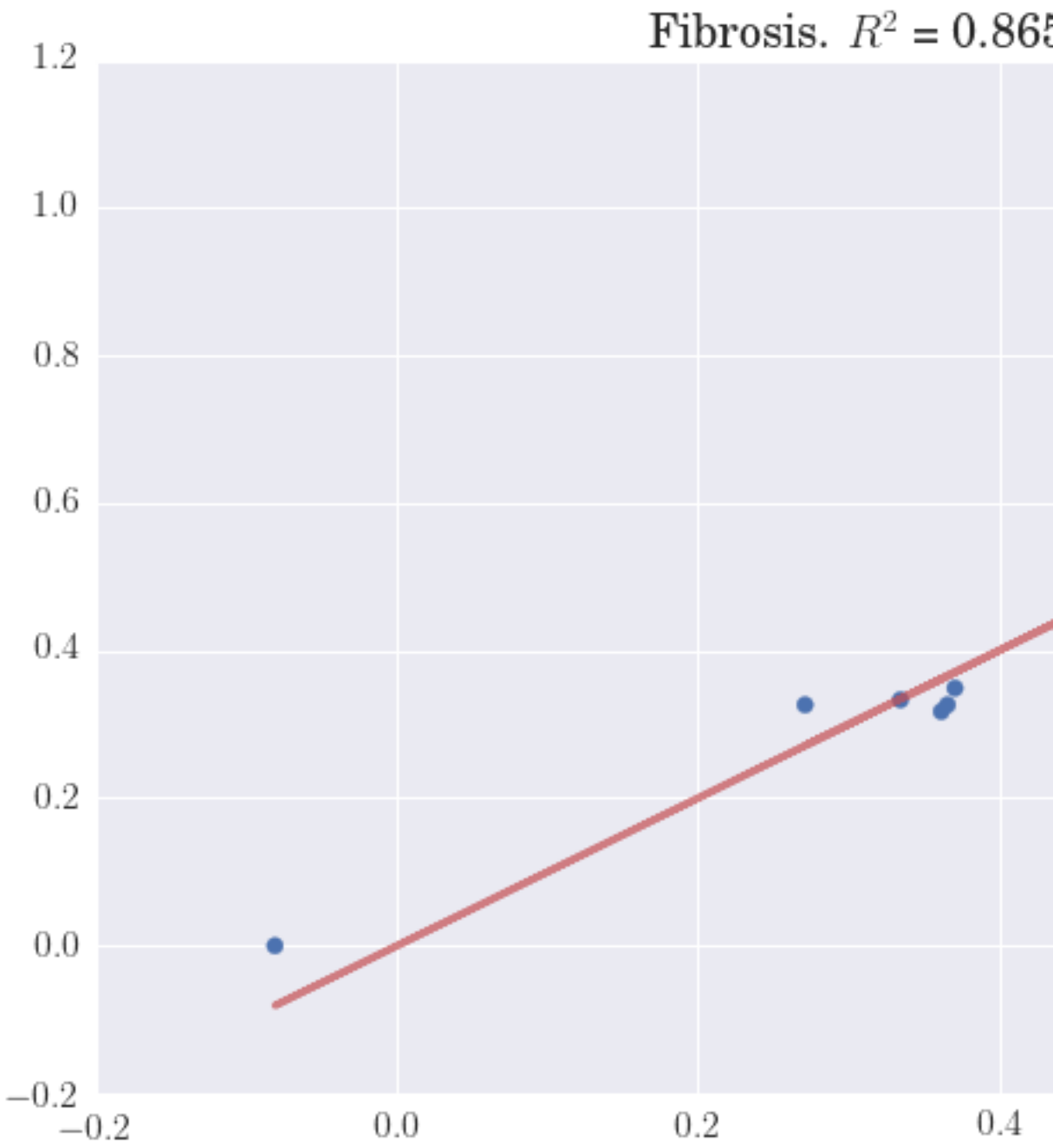


Figure 10: image

```
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
FociSize       1.1379      0.319      3.567      0.006         0.416     1.860
Intercept      0.0460      0.159      0.289      0.779        -0.314     0.406
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
"""
```

In [8]:
```
interface_hepatitis
```

Out[8]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            GLS Regression Results
==============================================================================
Dep. Variable:     Interface_hepatitis   R-squared:                       0.659
Model:                             GLS   Adj. R-squared:                  0.621
Method:                  Least Squares   F-statistic:                     17.38
Date:                 Wed, 29 Oct 2014   Prob (F-statistic):            0.00242
Time:                         11:13:48   Log-Likelihood:                 2.3063
No. Observations:                   11   AIC:                            -0.6126
Df Residuals:                        9   BIC:                             0.1832
Df Model:                            1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
Lacunarity    -1.0224      0.245     -4.168      0.002        -1.577    -0.468
Intercept      0.9504      0.143      6.669      0.000         0.628     1.273
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
"""
```

1.5 Exploratory analysis

1.5.1 with a random effect on age at death

Dependent variable	Models AIC < 2 + AICmin	Primary explanatory variables (ordered)
Ishak score	7	entropy, tissue-to-sinusoid, focus count, focus siz
Lobular collapse	5	entropy, lacunarity, tissue-to-sinusoid, focus coun
Confluent necrosis	1	entropy
Interface hepatitis	2	entropy, tissue-to-sinusoid
Portal inflammation	4	entropy, focus size, lacunarity, focus count, scale,
Fibrosis	2	entropy, lacunarity, tissue-to-sinusoid

Dependent variable	Models AIC $< 2 +$ AICmin	Primary explanatory variables (ordered)
Biliary hyperplasia	1	focus size
Necrosis, apoptosis, random inflammation	2	entropy, lacunarity

Measures we like

- entropy consistently explains histological measures when controlled for age
- also important : tissue to sinusoid ratio, focus count and size, lacunarity

1.6 Future directions

1.6.1 Further exploration of the dataset

- 145 sheep (89 females)
- 11 age classes
-
- 4460 entries across 27 variables
- 3330 with full image and histological information
- 1196 for which **complete** information is available

1.6.2 Narrow-field images

- 12536 images
- spatial distribution of nuclei

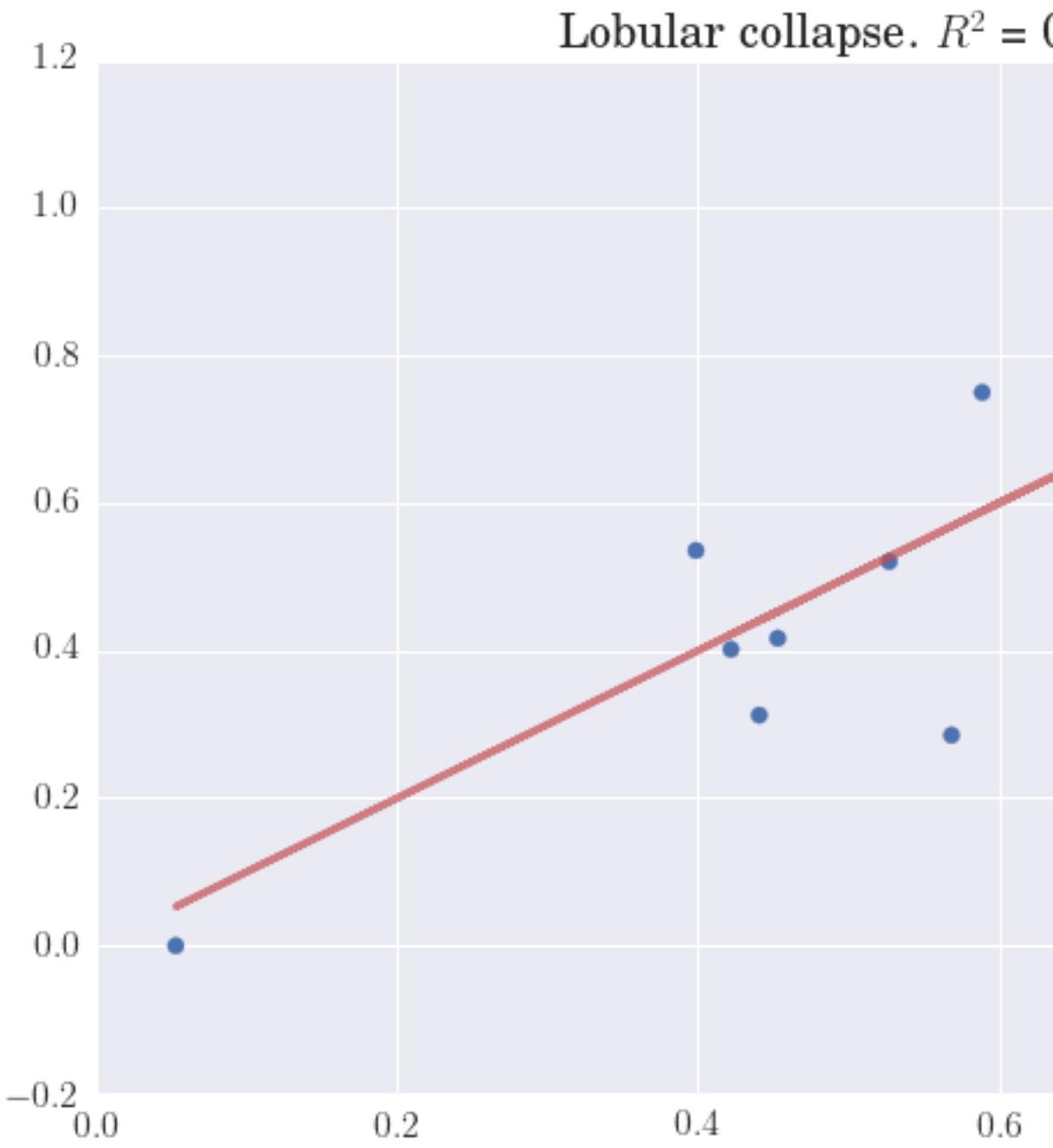


Figure 11: image

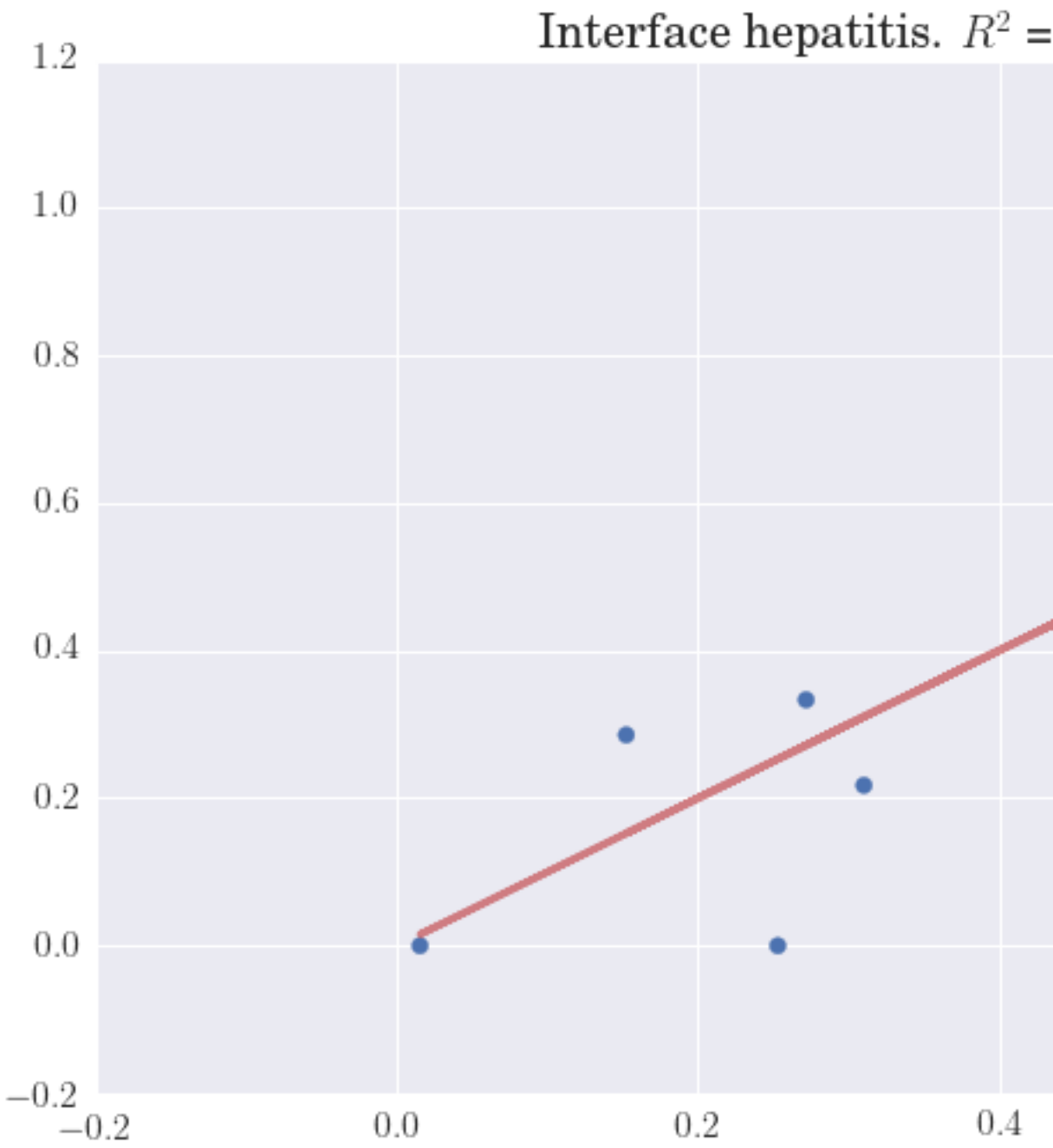


Figure 12: image