```
iter 118: ll=32254834.936543, converged: 100.00% (loc: 100.00%, scale update: False), in
0.34sec
iter 119: ll=32254834.936543, converged: 100.00% (loc: 100.00%, scale update: True), in
0.40sec
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
Input In [106], in <cell line: 1>()
----> 1 test_regressed_sf = de.test.wald(
      2     data=adata_human,
      3     gene_names=adata_human.var_names,
      4     formula_loc="~ 1 + palantir_pseudotime_int",
      5     factor_loc_totest="palantir_pseudotime_int",
      6     as_numeric=["palantir_pseudotime_int"]
      7 )

File ~/miniconda/envs/env_thesis7/lib/python3.10/site-packages/diffxpy/testing/tests.py:717, in
wald(data, factor_loc_totest, coef_to_test, formula_loc, formula_scale, as_numeric, init_a,
init_b, gene_names, sample_description, dmat_loc, dmat_scale, constraints_loc,
constraints_scale, noise_model, size_factors, batch_size, backend, train_args,
training_strategy, quick_scale, dtype, **kwargs)
    714 col_indices = np.array([np.where(constraints_loc_temp[x, :] == 1)[0][0] for x in
col_indices])
    716 # Fit model.
--> 717 model = _fit(
    718     noise_model=noise_model,
    719     data=data,
    720     design_loc=design_loc,
    721     design_scale=design_scale,
    722     design_loc_names=design_loc_names,
    723     design_scale_names=design_scale_names,
    724     constraints_loc=constraints_loc,
    725     constraints_scale=constraints_scale,
    726     init_a=init_a,
    727     init_b=init_b,
    728     gene_names=gene_names,
    729     size_factors=size_factors,
    730     batch_size=batch_size,
    731     backend=backend,
    732     train_args=train_args,
    733     training_strategy=training_strategy,
    734     quick_scale=quick_scale,
    735     dtype=dtype,
    736     **kwargs,
    737 )
```

```
        739 # Prepare differential expression test.
        740 de_test = DifferentialExpressionTestWald(
        741     model_estim=model,
        742     col_indices=col_indices,
        743     noise_model=noise_model,
        744     sample_description=sample_description
        745 )

File ~/miniconda/envs/env_thesis7/lib/python3.10/site-packages/diffxpy/testing/tests.py:248, in
_fit(noise_model, data, design_loc, design_scale, design_loc_names, design_scale_names,
constraints_loc, constraints_scale, init_model, init_a, init_b, gene_names, size_factors,
batch_size, backend, training_strategy, quick_scale, train_args, close_session, dtype)
        242 estim.train_sequence(
        243     training_strategy=training_strategy,
        244     **train_args
        245 )
        247 if close_session:
--> 248         estim.finalize()
        249 return estim

File
~/miniconda/envs/env_thesis7/lib/python3.10/site-packages/batchglm/train/numpy/base_glm/est
imator.py:544, in EstimatorGlm.finalize(self)
        536 """
        537 Evaluate all tensors that need to be exported from session and save these as class
attributes
        538 and close session.
   (...)
        541 transfers relevant attributes.
        542 """
        543 # Read from numpy-IRLS estimator specific model:
--> 544 self._hessian = - self.model.fim.compute()
        545 fisher_inv = np.zeros_like(self._hessian)
        546 invertible = np.where(np.linalg.cond(self._hessian, p=None) < 1 /
sys.float_info.epsilon)[0]

File
~/miniconda/envs/env_thesis7/lib/python3.10/site-packages/batchglm/train/numpy/base_glm/mo
del.py:136, in ModelIwls.fim(self)
        133 fim_ab = self.fim_ab
        134 fim_ba = np.transpose(fim_ab, axes=[0, 2, 1])
        135 return - np.concatenate([
--> 136         np.concatenate([fim_aa, fim_ab], axis=2),
        137     np.concatenate([fim_ba, fim_bb], axis=2)
        138 ], axis=1)

File <__array_function__ internals>:5, in concatenate(*args, **kwargs)

File ~/miniconda/envs/env_thesis7/lib/python3.10/site-packages/dask/array/core.py:1760, in
Array.__array_function__(self, func, types, args, kwargs)
  1757 if has_keyword(da_func, "like"):
  1758          kwargs["like"] = self
-> 1760 return da_func(*args, **kwargs)

File ~/miniconda/envs/env_thesis7/lib/python3.10/site-packages/dask/array/core.py:4200, in
concatenate(seq, axis, allow_unknown_chunksizes)
  4156 """
  4157 Concatenate arrays along an existing axis
  4158
  (...)
  4196 stack
  4197 """
  4198 from dask.array import wrap
-> 4200 seq = [asarray(a, allow_unknown_chunksizes=allow_unknown_chunksizes) for a in
seq]
  4202 if not seq:
  4203          raise ValueError("Need array(s) to concatenate")

File ~/miniconda/envs/env_thesis7/lib/python3.10/site-packages/dask/array/core.py:4200, in
<listcomp>(.0)
  4156 """
  4157 Concatenate arrays along an existing axis
  4158
  (...)
  4196 stack
  4197 """
  4198 from dask.array import wrap
-> 4200 seq = [asarray(a, allow_unknown_chunksizes=allow_unknown_chunksizes) for a in
seq]
  4202 if not seq:
  4203          raise ValueError("Need array(s) to concatenate")

File ~/miniconda/envs/env_thesis7/lib/python3.10/site-packages/dask/array/core.py:4548, in
asarray(a, allow_unknown_chunksizes, dtype, order, like, **kwargs)
  4546          else:
  4547          a = np.asarray(a, like=like_meta, dtype=dtype, order=order)
-> 4548 return from_array(a, getitem=getter_inline, **kwargs)
```

```
File ~/miniconda/envs/env_thesis7/lib/python3.10/site-packages/dask/array/core.py:3458, in
from_array(x, chunks, name, lock, asarray, fancy, getitem, meta, inline_array)
  3454          asarray = not hasattr(x, "__array_function__")
  3456 previous_chunks = getattr(x, "chunks", None)
-> 3458 chunks = normalize_chunks(
  3459          chunks, x.shape, dtype=x.dtype, previous_chunks=previous_chunks
  3460 )
  3462 if name in (None, True):
  3463          token = tokenize(x, chunks, lock, asarray, fancy, getitem, inline_array)

File ~/miniconda/envs/env_thesis7/lib/python3.10/site-packages/dask/array/core.py:3073, in
normalize_chunks(chunks, shape, limit, dtype, previous_chunks)
  3070 chunks = tuple("auto" if isinstance(c, str) and c != "auto" else c for c in chunks)
  3072 if any(c == "auto" for c in chunks):
-> 3073          chunks = auto_chunks(chunks, shape, limit, dtype, previous_chunks)
  3075 if shape is not None:
  3076          chunks = tuple(c if c not in {None, -1} else s for c, s in zip(chunks, shape))

File ~/miniconda/envs/env_thesis7/lib/python3.10/site-packages/dask/array/core.py:3248, in
auto_chunks(chunks, shape, limit, dtype, previous_chunks)
  3246          for i in small:
  3247          chunks[i] = (shape[i],)
-> 3248          return auto_chunks(chunks, shape, limit, dtype)
  3250 for i in autos:
  3251          chunks[i] = round_to(size, shape[i])

File ~/miniconda/envs/env_thesis7/lib/python3.10/site-packages/dask/array/core.py:3243, in
auto_chunks(chunks, shape, limit, dtype, previous_chunks)
  3239 if dtype.itemsize == 0:
  3240          raise ValueError(
  3241          "auto-chunking with dtype.itemsize == 0 is not supported, please pass in
`chunks` explicitly"
  3242          )
-> 3243 size = (limit / dtype.itemsize / largest_block) ** (1 / len(autos))
  3244 small = [i for i in autos if shape[i] < size]
  3245 if small:

ZeroDivisionError: float division by zero
```